

Teamcenter 10.1

System Administration Guide

Proprietary and restricted rights notice

This software and related documentation are proprietary to Siemens Product Lifecycle Management Software Inc.

© 2013 Siemens Product Lifecycle Management Software Inc. All Rights Reserved.

Siemens and the Siemens logo are registered trademarks of Siemens AG. Teamcenter is a trademark or registered trademark of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other trademarks, registered trademarks, or service marks belong to their respective holders.

Contents

Proprietary and restricted rights notice	2
Getting started with system administration	1-1
Introduction to system administration	1-1
Prerequisites	1-1
Teamcenter applications used for system administration	1-1
Syntax definitions	1-2
Basic concepts	1-3
Basic tasks	1-12
Process daemons	2-1
Introduction to process daemons	2-1
Action Manager daemon	2-1
ODS and IDSM daemons	2-3
Subscription Manager daemon	2-3
Task Manager daemon	2-4
Encrypting a password file for use by daemons	2-5
Maintaining the database server	3-1
Maintaining the IBM DB2 server	3-1
Moving a DB2 database from Windows to Linux	3-3
Maintaining the Oracle database	3-4
Oracle Net implementation	3-27
Maintaining the Microsoft SQL server database	3-29
Implementing Microsoft SQL Net	3-31
Teamcenter licenses	4-1
Named user licensing	4-1
Common licensing server	4-2
Managing licenses	4-2
Using the convert_license_log utility	4-3
Generating license usage and module usage reports	4-4
Using the LicenseUsedAuditTool tool	4-6
Teamcenter client communication system (TCCS)	5-1
Managing Teamcenter client communication system (TCCS)	5-1
TCCS configuration files	5-3
Configuring multiple TCCS environments	5-8
TCCS and container applications	5-9
Administering proxy support for clients not integrated with TCCS	5-12
TCCS logging	5-13
Server manager	6-1

Contents

Introduction to the server manager	6-1
Server manager prerequisites	6-2
Server manager properties files	6-3
Monitoring	6-11
Server manager logging	6-27
J2EE server manager administrative interface	6-32
.NET server manager administrative interface	6-40
Using third-party applications to view server manager administration data	6-45
Updating property values in bulk	7-1
Process for updating property values in bulk	7-1
Using command line arguments to update property values in bulk	7-2
Using an XML input file to update property values in bulk	7-5
Importing updated properties back into Teamcenter	7-7
Performance statistics	7-7
Best practices and considerations	7-8
File Management System	8-1
Introduction to File Management System	8-1
Benefits of using FMS	8-1
FMS components	8-3
FMS configuration files	8-5
Sizing FMS fast cache	8-18
Administering FMS	8-45
Improving cache performance	8-132
Sample FMS configurations	8-132
Configuring Teamcenter for performance	9-1
Configuring the four-tier architecture for performance	9-1
Configuring the rich client for startup performance	9-18
Cleaning the POM_timestamp table	9-26
Cleaning the backpointer table after upgrade	9-26
Logging	10-1
Introduction to logging	10-1
Using the Log Manager	10-1
Logging for business logic servers	10-3
Logging for Teamcenter tiers	10-5
Backing up and recovering files	11-1
Overview of the backup and recovery process	11-1
Oracle Recovery Manager (RMAN)	11-3
Restoring purged files	11-7
Using alternative hot backup and recovery procedures	11-9
Glossary	A-1
Index	Index-1

Figures

Two-tier architecture	1-3
Two-tier deployment	1-4
Four-tier architecture	1-6
Four-tier deployment (enterprise and Web tiers on same host)	1-8
Four-tier deployment (enterprise and Web tiers on separate hosts with HTTP server)	1-9
Four-tier deployment (multiple enterprise tier hosts and Web tier hosts)	1-10
Four-tier deployment (load balancing)	1-11
Password Security panel in TEM	2-5
License Usage Report output	4-5
Global configuration	6-44
Teamcenter server	6-45
Installing the StoreAndForward translator	8-91
Running the store_and_forward translator in the Dispatcher Admin Client	8-93

Chapter

1 Getting started with system administration

Introduction to system administration	1-1
Prerequisites	1-1
Teamcenter applications used for system administration	1-1
Syntax definitions	1-2
Basic concepts	1-3
Teamcenter architecture	1-3
Two-tier architecture	1-3
Four-tier architecture	1-4
Database server	1-12
Teamcenter clients	1-12
Rich client	1-12
Thin client	1-12
Basic tasks	1-12
File management	1-12
Backing up and recovering Teamcenter files	1-13
Tuning the four-tier architecture for performance	1-13
Finding error codes	1-13

Chapter

1 Getting started with system administration

Introduction to system administration

This guide is for persons responsible for configuring and managing system and database elements. It describes the basic concepts behind Teamcenter database administration and includes information about how to choose architectural elements, back up or recover files, manage volumes, tune the system for performance, and access log files.

This guide is intended for Teamcenter administrators who are responsible for:

- Determining Teamcenter architecture (two-tier or four-tier)
- Managing database elements (Oracle or MS SQL Server tools)
- Configuring Teamcenter servers and clients
- Managing Teamcenter files and volumes
- Backing up and recovering Teamcenter files
- Tuning the four-tier architecture for performance
- Using log files to troubleshoot system performance

It is assumed that the administrator is familiar with the Teamcenter architecture and its installation, and with third-party database elements.

Prerequisites

You must have administrative privileges to perform most system and database tasks.

For additional information about defining and using administrative privileges, see the [Organization Guide](#).

Teamcenter applications used for system administration

Some system administration tasks can be performed from the rich client interface. Use the following Teamcenter applications to manage FMS volumes.

Application	Description
Organization	<p>Create and modify volumes and their properties. You can also create and maintain your company's virtual organization within Teamcenter. Use this digital representation of your company to manage user accounts, group accounts, and their respective permissions.</p> <p>For detailed information about this application, see the Organization Guide.</p>
Volume Management	<p>Migrate infrequently used files from a source volume to a destination volume, without using third-party storage systems. Create and manage migration policies for both hierarchical storage management and volume management storage methods.</p> <p>For detailed information about this application, see the Volume Management Guide.</p>

Syntax definitions

This manual uses a set of conventions to define the syntax of Teamcenter commands, functions, and properties. Following is a sample syntax format:

harvester_jt.pl [*bookmark-file-name bookmark-file-name ...*]
[*directory-name directory-name ...*]

The conventions are:

Bold	Bold text represents words and symbols you must enter exactly as shown.
	In the preceding example, you enter harvester_jt.pl exactly as shown.
<i>Italic</i>	Italic text represents values that you supply.
	In the preceding example, you supply values for <i>bookmark-file-name</i> and <i>directory-name</i> .
<i>text-text</i>	A hyphen separates two words that describe a single value.
	In the preceding example, <i>bookmark-file-name</i> is a single value.
[]	Brackets represent optional elements.
...	An ellipsis indicates that you can repeat the preceding element.

Following are examples of correct syntax for the **harvester_jt.pl**: command, derived from conventions used in the documentation:

```
harvester_jt.pl
harvester_jt.pl assembly123.bkm
harvester_jt.pl assembly123.bkm assembly124.bkm assembly125.bkm
harvester_jt.pl AssemblyBookmarks
```

Basic concepts

Teamcenter architecture

Two-tier architecture

The two-tier architectural model comprises the following tiers:

- Client tier

The client tier comprises the Teamcenter rich clients.

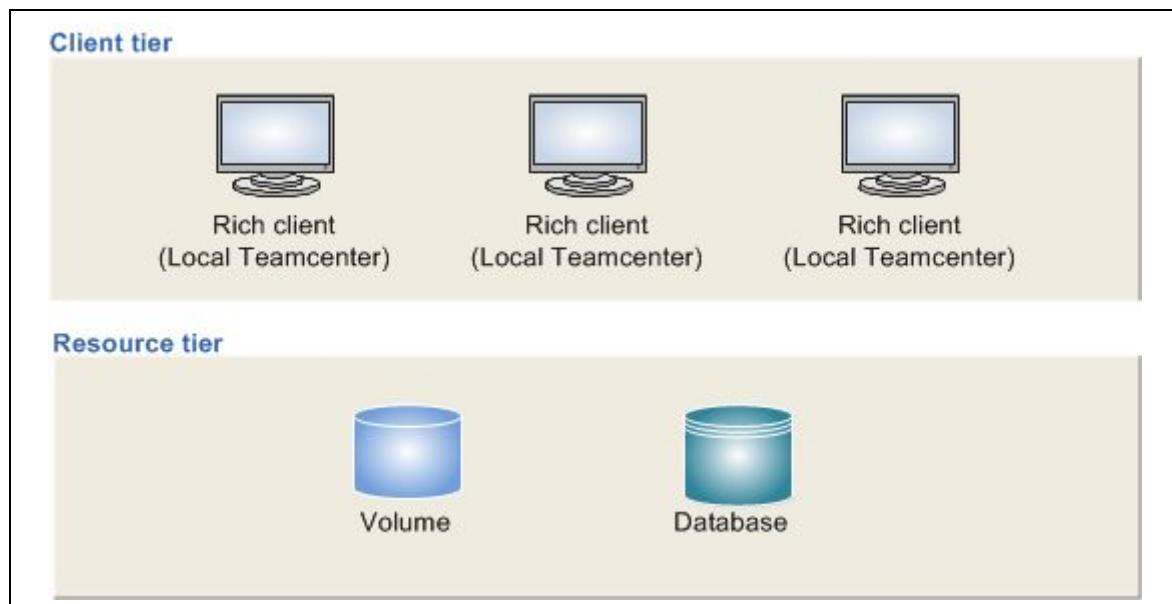
In a deployment of the two-tier architecture, the Teamcenter server runs on the client workstation.

Note The two-tier rich client is installed only through TEM. Over-the-Web installation is supported only for the four-tier rich client.

Some Teamcenter client features, such as Teamcenter Integration for NX, Lifecycle Visualization, and Teamcenter Client for Microsoft Office, require the Web tier, a component of the four-tier architecture. To enable these features for a two-tier rich client, you can connect the two-tier rich client to a deployment of the Web tier. For information about functionality you can add to a rich client and which add-ons require the Web tier, see the appropriate server installation guide (for [Windows](#) or [UNIX/Linux](#)).

- Resource tier

The resource tier comprises a database server, database, volumes, and file servers.



Two-tier architecture

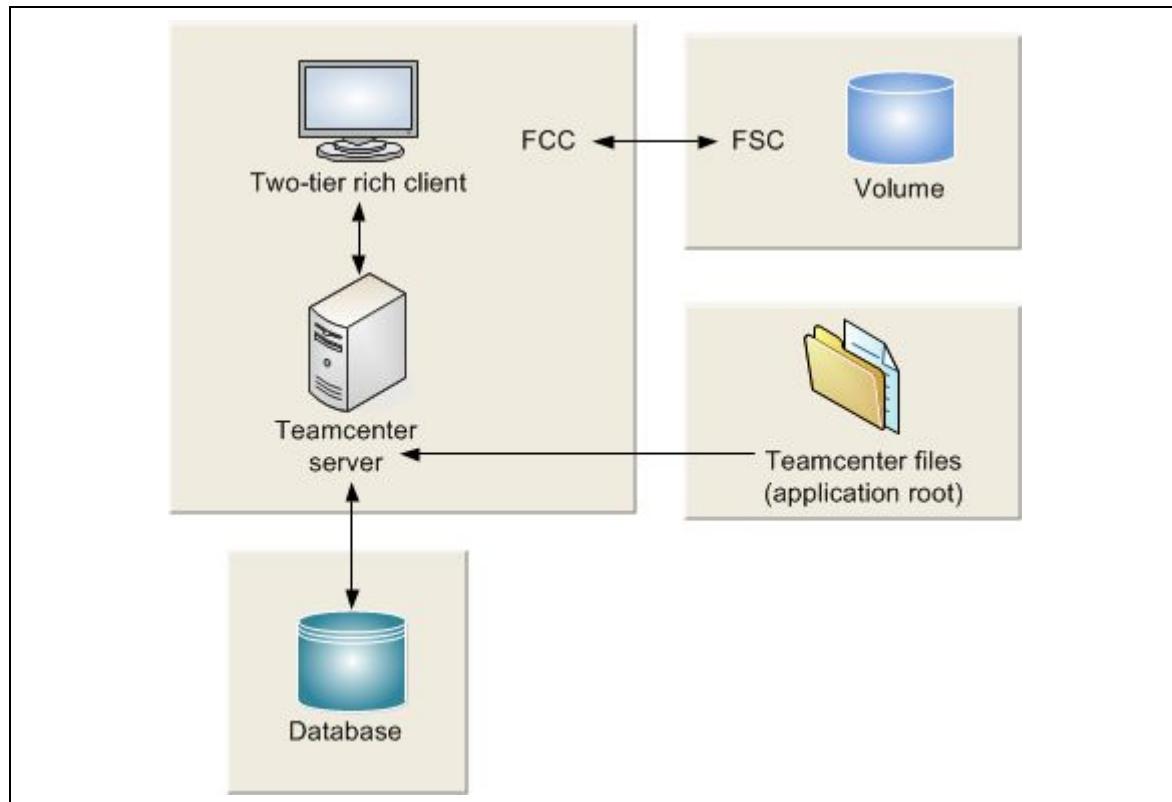
In the two-tier model, you deploy the Teamcenter rich client, which includes the local server, and the optional applications that integrate with the rich client on the

client workstation. Typically, the database server, volumes, and file servers are installed on one or more separate hosts.

Teamcenter File Management System (FMS) manages the rich client access to volumes:

- The FMS server cache (FSC) process run on the server hosting the volume.
- The FMS client cache (FCC) process runs on the rich client host.

For more information about FMS and two-tier rich client architecture, see the appropriate Teamcenter server installation guide (for [Windows](#) or [UNIX/Linux](#)).



Two-tier deployment

Four-tier architecture

The four-tier architecture model comprises the following tiers:

- Client tier

The client tier comprises the Teamcenter rich client, thin client, and other clients such as Teamcenter Client for Microsoft Office.

Note The rich client can be deployed with additional functionality, such as Lifecycle Visualization, Teamcenter Client for Microsoft Office, and Teamcenter Integration for NX or NX Integration 4.0.1. (Teamcenter Integration for NX/NX Integration 3 is not supported.)

- J2EE Web tier

The J2EE Web tier is a Java application that runs in a Java 2 Enterprise Edition (J2EE) application server, such as Oracle WebLogic, and is responsible for communication between the client tier and enterprise tier. For information about supported application servers, see the Siemens PLM Software Certification Database:

[http://support.industrysoftware.automation.siemens.com/
certification/teamcenter.shtml](http://support.industrysoftware.automation.siemens.com/certification/teamcenter.shtml)

- Enterprise tier

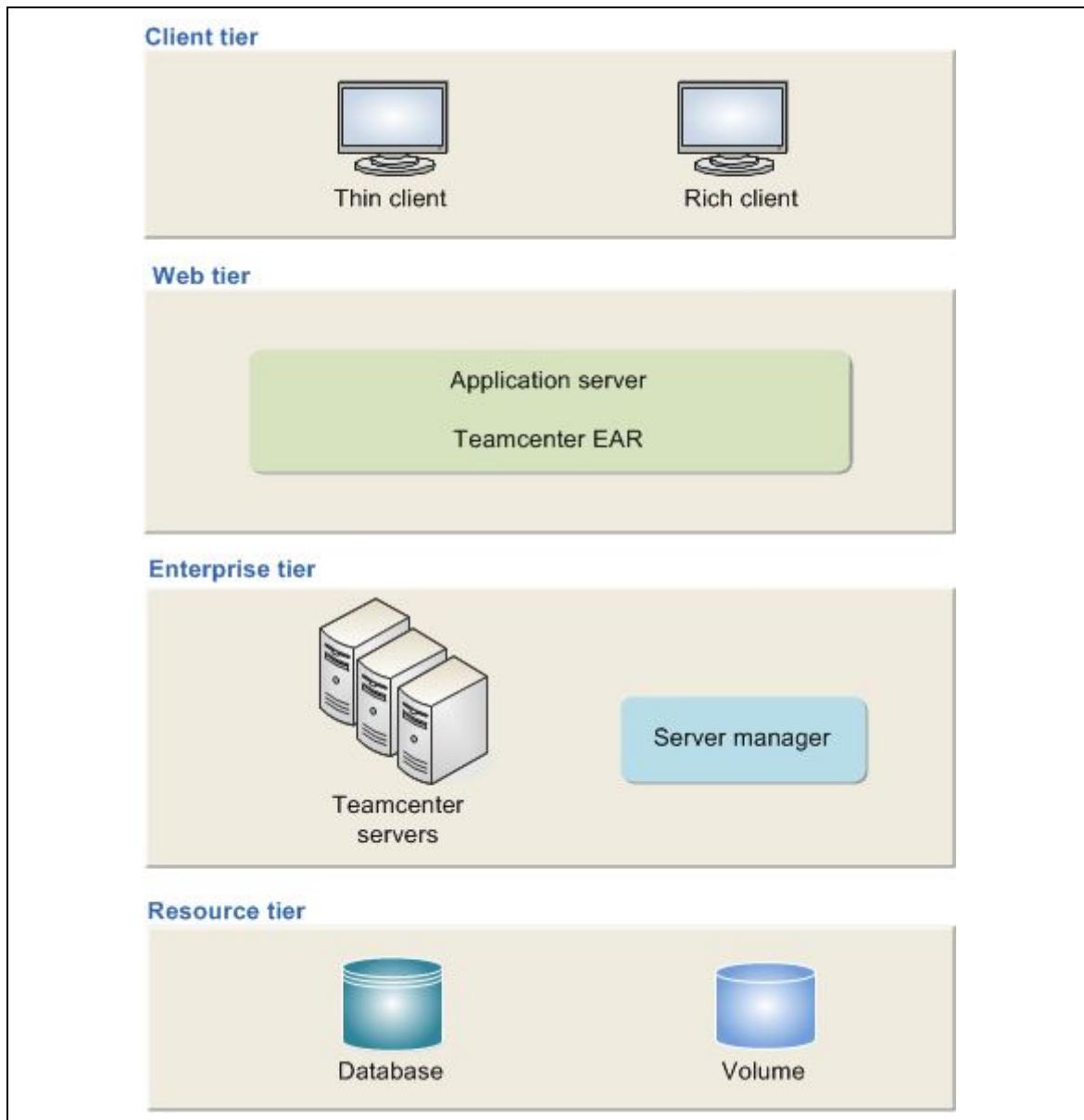
The enterprise tier comprises a configurable pool of Teamcenter C++ server processes and a server manager. The enterprise tier retrieves data from and stores data in the database.

A server manager manages a pool of Teamcenter server processes. You must install a server manager whenever you deploy the Web tier.

Note Teamcenter provides server managers based on the J2EE and the Microsoft .NET platforms. Install the appropriate server manager for the Web tier you use. The .NET Web tier is supported only on Windows platforms.

- Resource tier

The resource tier comprises a database server, database, volumes, and file servers.



Four-tier architecture

You can design deployments that host the Web tier, resource tier, and enterprise tiers on the same computer or on separate computers:

- Smaller sites can run the pool of servers and the server manager on the same host as the Web tier.
- Larger sites can distribute the pool of server processes across multiple hosts and optionally include an HTTP server to serve static files or multiple HTTP servers to support load balancing.

For a multihost configuration, the server pool consists of multiple subpools, one or more for each host. Each subpool is managed by one server manager process. The Web tier balances the load across the server pools.

The Teamcenter J2EE based server manager and Web tier application both employ the JBoss cache, a tree-structured cache, to provide replication and transaction context. You must configure the JBoss cache (called *TreeCache* in Teamcenter) in both the J2EE based server manager and the Web tier application.¹

To ensure communication between the Web tier and the server manager, you must coordinate the values you specify for each component. For some values, you must provide the identical value when configuring the Web tier application.

If you are setting up multiple Web tier environments with separate domains, you must configure:

- A minimum of one server manager for each Web tier deployment.
- A separate TreeCache cluster for each environment.

To configure a separate TreeCache cluster, Siemens PLM Software recommends configuring a different port (multicast) or set of ports (TCP) for each cluster.

The JMX HTTP adapter allows you to view the status of the server pool and dynamically alter the pool configuration values (the values are not persistent). Access this functionality from the following URL:

http://host-name:jmx-port

Replace *host-name* with the name of the host running the server manager. Replace *jmx-port* with the number of the port running the JMX HTTP adapter. This port number is defined when you install the J2EE based server manager.

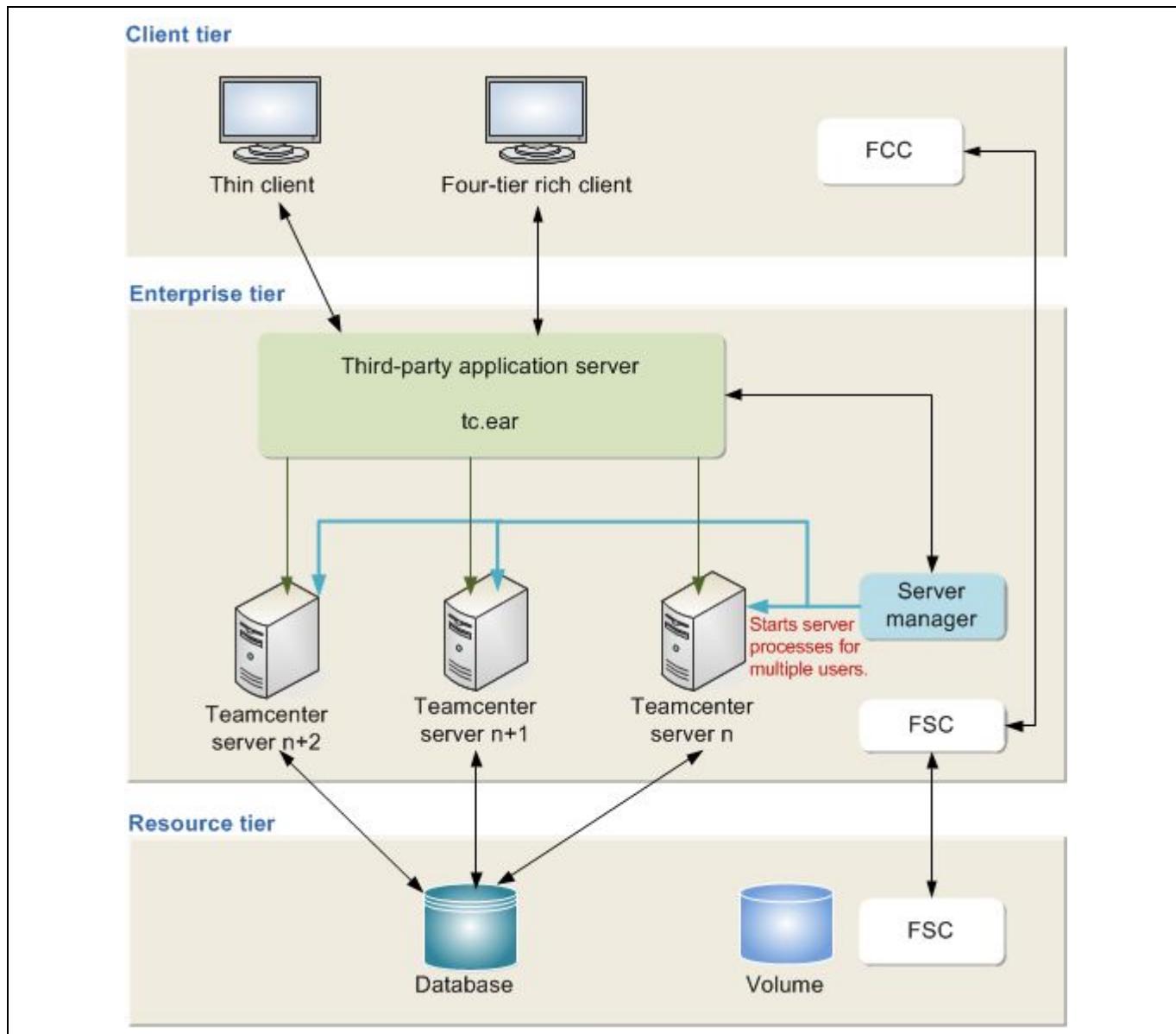
The first time you log on to the adapter, use **manager** for both the user name and the password. You can change the user name and password to unique values using the adapter.

Teamcenter File Management System (FMS) manages the rich client access to volumes:

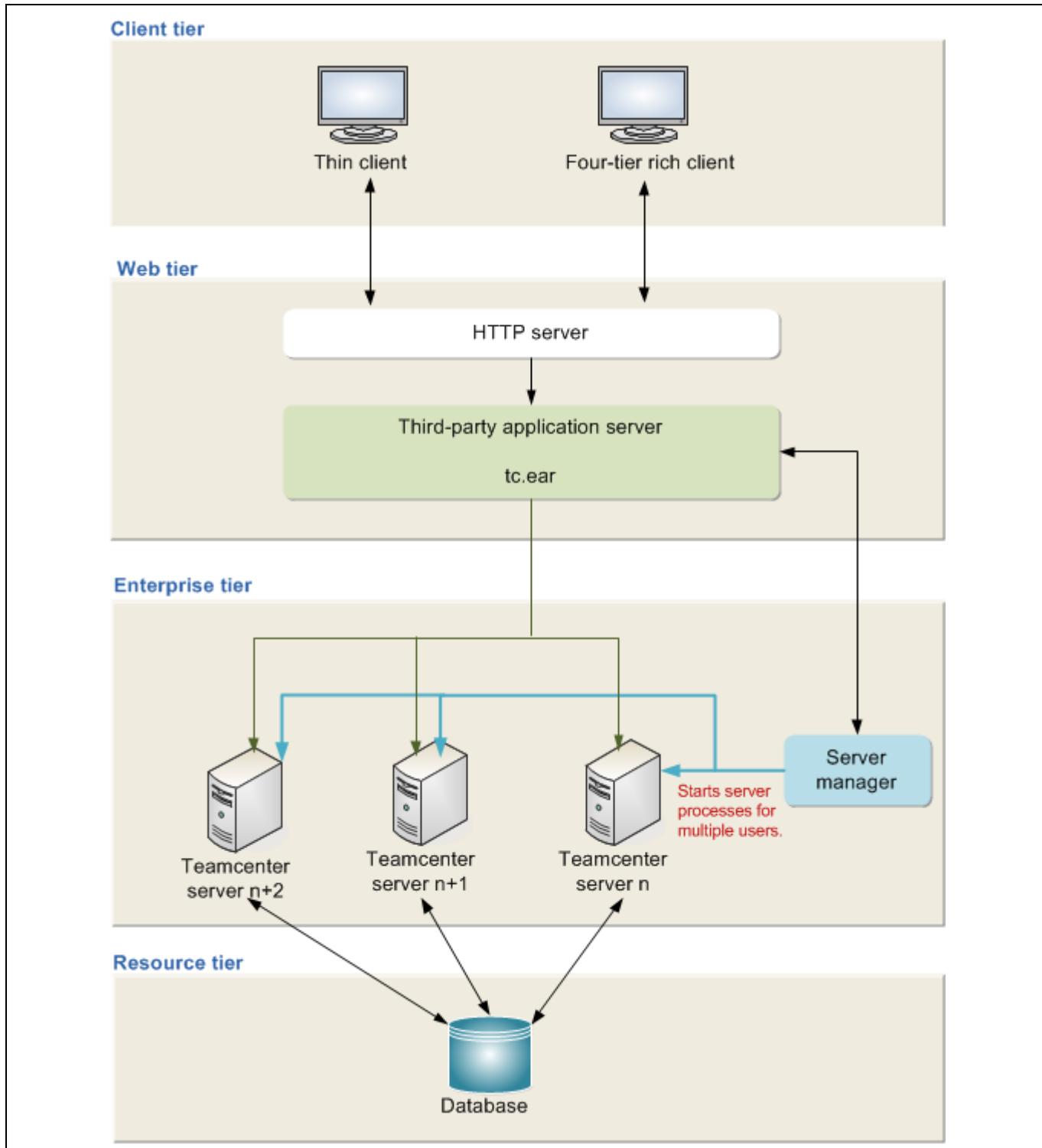
- The FMS client cache (FCC) process runs on the rich client host.
- The FMS server cache (FSC) process runs on each server hosting a volume and each server hosting a pool of Teamcenter servers (**TcServer**).

Note If you install File Management System, the FMS server cache (FSC) and the server manager must run on the same host server, with the same user ID.

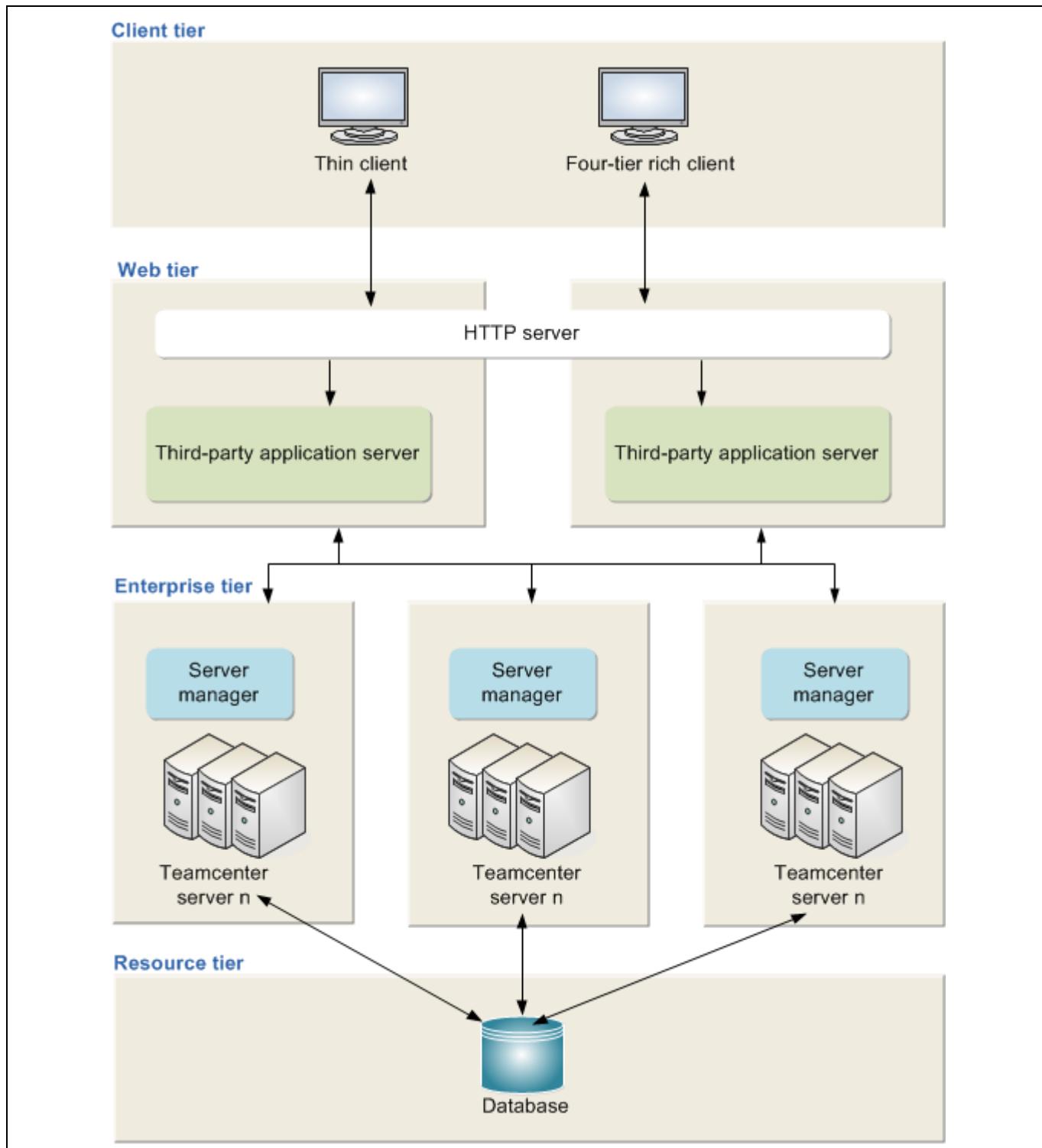
1. This is not required if you use the .NET Web tier and the .NET based server manager.



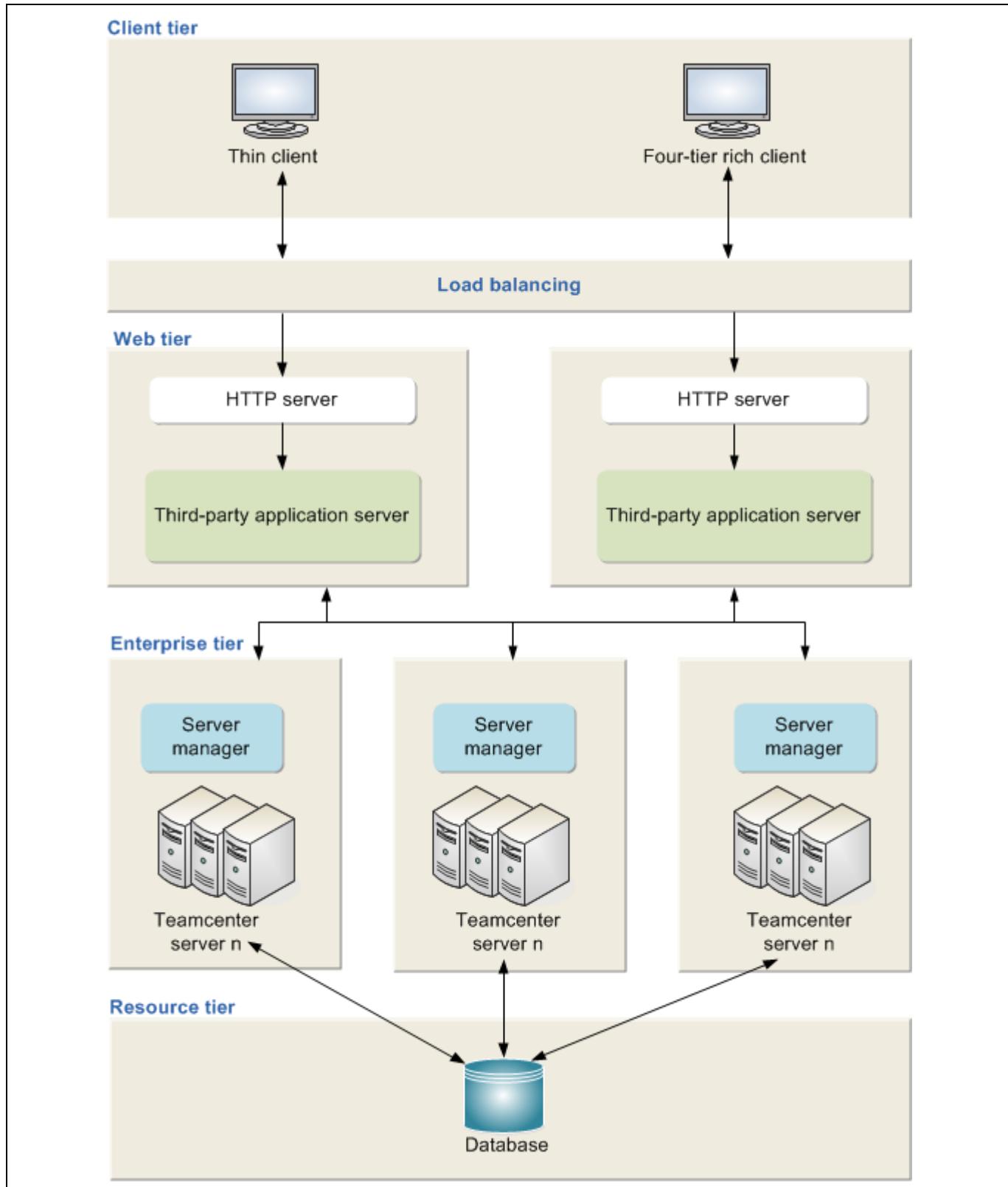
Four-tier deployment (enterprise and Web tiers on same host)



Four-tier deployment (enterprise and Web tiers on separate hosts with HTTP server)



Four-tier deployment (multiple enterprise tier hosts and Web tier hosts)



Four-tier deployment (load balancing)

Database server

A Teamcenter network requires access to a database server.

Before you install Teamcenter, you or your database administrator must install and configure a database server to store Teamcenter data. The Teamcenter corporate server must have access to a database server or a database client. Teamcenter supports IBM DB2, Oracle, and Microsoft SQL Server databases.

For Oracle configuration settings and tuning methods to optimize Teamcenter performance, see the [Teamcenter Deployment Guide](#), available in the documentation section of Siemens PLM Software's support site. The [Teamcenter Deployment Guide](#) also provides an in-depth review of Oracle database performance issues and diagnosis, and configuration and tuning guidelines.

For Microsoft SQL Server configuration settings and tuning methods to optimize Teamcenter performance, see the [Teamcenter Deployment Guide](#), available in the documentation section of Siemens PLM Software's support site. The [Teamcenter Deployment Guide](#) also provides an in-depth review of Microsoft SQL database performance issues and diagnosis, and configuration and tuning guidelines.

Note Teamcenter servers and two-tier rich clients on Linux hosts cannot connect to Microsoft SQL Server database servers.

Teamcenter clients

Rich client

The *rich client* is a platform-independent client implementation (Java application) for users who interact with Teamcenter frequently. It is extendable and able to run both Teamcenter and customer-written applications. Customers can also extend the standard user interface.

The rich client application is deployed on each user workstation using Teamcenter Environment Manager or the Over-the-Web Installer, depending on which Teamcenter network architecture you use. The rich client is supported in both architectural models described in [Two-tier architecture](#) and [Four-tier architecture](#).

Thin client

The *thin client* provides access to Teamcenter through a standard commercial Web browser, such as Microsoft Internet Explorer or Mozilla Firefox. The user interface provides a streamlined browser-based view of product information stored in a Teamcenter database.

The thin client is supported only in the four-tier architectural model described in [Four-tier architecture](#).

Basic tasks

File management

An initial install of Teamcenter using Teamcenter Environment Manager provides a single FMS server cache (FSC) that mounts to a single volume; a typical

configuration for small deployment. During installation, Teamcenter Environment Manager 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 has been installed, create and modify volumes using the Organization application. For detailed information about managing volumes, see the [Organization Guide](#).

You can also customize your FMS configuration after the initial installation. For additional information, see [Manually configuring an FSC](#). For sample configuration examples, see [About the sample FMS configurations](#).

Backing up and recovering Teamcenter files

Teamcenter's integrated backup and recovery feature facilitates third-party backup systems to perform online backup, allowing Teamcenter to operate continually. This functionality focuses on the area of backing up metadata and math data, and recovering that data in different restoration scenarios.

For more information about backing up and recovering files, see [Overview of the backup and recovery process](#).

Tuning the four-tier architecture for performance

You can tune Teamcenter's four-tier architecture for optimum performance. The default settings for application servers are rarely appropriate for production scalability or good transactional performance.

For more information about tuning for performance, see [Introduction to configuring the four-tier architecture for performance](#).

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*.

Note The *Integration Toolkit Function Reference* is available only in the Teamcenter HTML Help Collection. It is not available in the PDF collection.

2. At the top of the page, select the **Modules** header.

3. 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 **EPM Errors** and click the link.

4. The error page displays all errors for that module. Error numbers are defined as *module base value + error code*.

For example, the **EPM_internal_error** error has an error code of **EMH_EPM_error_base + 1**.

5. To determine the error base value for the selected module:
 - a. Return to the **Modules** page.
 - b. Scroll down to **EMH Constants** and click the link.
 - c. 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**.

Thus, the error number for the **EPM_internal_error** error is the concatenation of the EPM modules error base (**33000**) and the error code (**1**), creating an error code of **33001**.

Chapter

2 *Process daemons*

Introduction to process daemons	2-1
Action Manager daemon	2-1
ODS and IDSM daemons	2-3
Subscription Manager daemon	2-3
Task Manager daemon	2-4
Encrypting a password file for use by daemons	2-5

Chapter

2 *Process daemons*

Introduction to 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 users' inboxes for overdue tasks.

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.

For more information, see [Encrypting a password file for use by daemons](#).

You can install process daemons from Teamcenter Environment Manager (TEM) during installation or maintenance.

For more information about using TEM to install process daemons, see the [Teamcenter Environment Manager Help](#).

Action Manager daemon

actionmgrd [-help] [-u=username -p=password -g=group] [&]

-u

Specifies the user ID.

-p

Specifies the password.

-g

Specifies the group associated with the user.

&

Executes the command asynchronously in a subshell. The shell does not wait for the command to finish before executing the next command.

Dispatches events given a deferred execution time at the time specified, as well as events intended for immediate dispatch that failed to execute immediately. Information regarding events with a deferred execution time are stored in the Teamcenter database as *action objects*. The action object contains the information necessary to process the event, including event type, execution time, and the user who generated the event.

This daemon also processes events requiring a workflow action handler to run a defined subprocess. The handler is executed by either calling its associated handler

function (allowing the handler to run within the context of the daemon's process) or by initiating a separate subprocess. The method used is determined by how the handler is configured in the database.

More than one action manager process daemon can be run at your site. Additional daemons can be run on the same node, or on a different node. If you anticipate many events, Siemens PLM Software recommends you run the daemons on server nodes that users do not directly log on to improve performance.

Note Automatic logon is recommended for security reasons. Only a system administrator can start this daemon.

At defined intervals, the daemon searches for action objects stored in the Teamcenter database. It determines which of the objects are due for execution and processes a defined number of these objects. To avoid memory leak problems, the daemon clones and then terminates itself on a regular basis.

The duration of its sleep and wake cycles, the number of objects processed and its cloning intervals are determined by the following preferences:

- **TC_actionmgrd_sleep_minutes**

Defines the number of minutes in the sleep period of the **actionmgrd** process daemon between dispatching cycles.

- **TC_actionmgrd_general_processing_hours**

Allows you to define processing periods for the **actionmgrd** process daemon at specified times during a 24-hour day. Use this preference as an alternative to the **TC_actionmgrd_sleep_minutes** preference, which is based upon a continuous sleep/dispatch cycle.

- **TC_actionmgrd_max_actions_to_dispatch**

Defines how many action events are dispatched every time the **actionmgrd** process daemon processes actions events in the queue. Only the action events determined to be ready for execution are processed.

- **TC_actionmgrd_max_subprocess_to_start**

Defines how many events the **actionmgrd** process daemon dispatches by creating a separate subprocess run by a workflow action handler. Once the defined number is exceeded, the daemon does not dispatch any event requiring an action handler to execute as a separate subprocess.

- **TC_actionmgrd_cloning_interval**

Defines the number of cycles the **actionmgrd** process daemon completes before cloning and terminating itself to avoid memory leak problems.

When running multiple **actionmgrd** process daemons, the operation of each daemon can be independently controlled by using separate preference files in separate **TC_DATA** directories.

ODS and IDSM daemons

Both the Object Directory Service (ODS) and the Integrated Directory Services Manager (IDSM) require a server process or daemon. The network nodes that run these daemons are referred to as the ODS or IDSM server nodes, respectively.

The ODS daemon is started by the **run_tc_ods** script and runs until the process is stopped or the ODS server node is shut down. There is only one ODS daemon per ODS and it automatically connects to the ODS database using the **infodba** user account.

The IDSM daemon is dynamically started using the **run_tc_idsm** script and runs until it has accomplished its task of transporting a set of objects from one site to another. It then transitions to a dormant state for approximately two minutes, then terminates if it is not reused for another request.

There can be more than one IDSM daemon running on the same IDSM server node at a given time. In fact, there will be one IDSM daemon for every Multi-Site Collaboration request to deliver an object. This is an important factor to consider when configuring an IDSM server node.

Each IDSM daemon automatically logs in to the working site database that it serves using the **infodba** user account. For sites using rules-based object protection, it is recommended that this user account be changed to a special account (for example, IDSM) so that the IDSM daemon runs under the context of a user that can be controlled. This technique makes it possible to define rules based on the IDSM user account for maximum security.

Subscription Manager daemon

subscripmgrd [-help] [-u=] *username* -p=*password* -g=*group*] [&]

-u

Specifies the user ID.

-p

Specifies the password.

-g

Specifies the group associated with the user.

&

Executes the command asynchronously in a subshell. The shell does not wait for the command to finish before executing the next command.

Searches the Teamcenter database for subscriptions to the object for a given event type. When a subscription event occurs (such as when an object is released) an *event object* is created to capture that event and stored in a queue in the database. Event types act as the catalyst for the **subscripmgrd** daemon, causing it to search for subscriptions to the object for the given event type.

When matching subscriptions are found, the daemon determines if the action handler associated with the subscription is executed immediately or if it has a deferred execution time. For immediate events, the daemon calls the handler's associated function, thereby executing the handler in the context of the daemon's

process. For deferred events, the daemon creates an action object for each deferred subscription. Action objects are processed by the action manager daemon.

For more information, see [Action Manager daemon](#).

If an error occurs when the daemon executes the subscription's action handler, it creates an action object, allowing the **actionmgrd** daemon to retry the handler. If a subscription handler is defined in the database requiring a separate process, the daemon creates an action object, deferring the execution to the **actionmgrd** daemon, which will create a separate subprocess to execute the handler.

Note Automatic logon is recommended for security reasons. Only a system administrator can start this daemon.

At defined intervals, the daemon searches for event objects stored in the Teamcenter database. It dispatches each event by searching for subscriptions that match the information in the even object. To avoid memory leak problems, the daemon clones and then terminates itself on a regular basis.

The duration of its sleep and wake cycles, the number of subscriptions dispatched and its cloning intervals are determined by the following preferences:

- **TC_subscriptionmgrd_sleep_minutes**

Defines the number of minutes in the sleep period of the **subscripmgrd** process daemon between dispatching cycles.

- **TC_subscriptionmgrd_processing_hours**

Allows you to define processing periods for the **subscripmgrd** process daemon at specified times during a 24-hour day. Use this preference as an alternative to the **TC_subscriptionmgrd_sleep_minutes** preference, which is based upon a continuous sleep/dispatch cycle.

- **TC_subscriptionmgrd_max_subscriptions_to_dispatch**

Defines how many subscriptions are dispatched every time the **subscripmgrd** process daemon processes subscription objects.

- **TC_subscriptionmgrd_cloning_interval**

Defines the number of cycles the **subscripmgrd** process daemon completes before cloning and terminating itself to avoid memory leak problems.

When running multiple **subscripmgrd** process daemons, the operation of each daemon can be independently controlled by using separate preference files in separate *TC_DATA* directories.

Task Manager daemon

The **Task Manager** daemon 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\log\taskmonitor_graceful_exit.tmp >>*.

Encrypting a password file for use by daemons

Teamcenter stores passwords in disk files using advanced encryption standard (AES) 256-bit encryption. These encrypted passwords are used by Teamcenter utilities, services (daemons) that access the Teamcenter database, and Teamcenter Environment Manager (TEM).

During installation, the TEM installer provides a **Password Security** panel that allows you to designate the path to the directory that contains the password files. TEM also locks access to the password directory to all users except the operating system user performing the installation. In maintenance mode, choose **Update Security** in the **Feature Maintenance** panel in TEM to change the directory where the password files are stored. Changing the stored password does not change the Teamcenter database password. This must be done separately in the rich client interface.

Caution For security reasons, access to the password directory must not be changed unless other methods of ensuring access control are in place.



Password Security panel in TEM

To create an encrypted password file, use the **install** utility with the **-encryptpwf** argument.

You can change the encryption key used to encode the password by creating a **CryptKey** file in the **TC_DATA** directory and providing the key in the file. If this **CryptKey** file exists and contains a valid key (32 bytes or more), this key is used instead of the key from the database.

Note Changing the encryption key is not required and not a common practice.

For more information about password management for utilities, see the [Utilities Reference](#). For more information about using encryption when changing the Oracle password, see the server installation guides ([Windows](#) or [UNIX/Linux](#)).

Chapter

3 *Maintaining the database server*

Maintaining the IBM DB2 server	3-1
Tune the IBM DB2 server	3-1
Start the DB2 service	3-1
Stop the DB2 service	3-2
Open the DB2 Control Center	3-2
Verify database connectivity	3-2
Moving a DB2 database from Windows to Linux	3-3
 Maintaining the Oracle database	3-4
Tune the Oracle database	3-4
Set the Oracle environment	3-5
Manually set the Oracle environment	3-5
Manually set the PATH environment variable	3-6
Manually set the shared library path (UNIX systems)	3-6
Oracle services (Windows systems)	3-6
Introduction to Oracle services for Windows systems	3-6
Manually start Oracle services	3-7
Before manually starting Oracle Services	3-7
Start the Oracle listener	3-7
Start database services	3-8
Initialize a database instance using SQL*Plus Utility	3-8
Manually stop Oracle services	3-9
Stop the Oracle listener	3-9
Stop database services	3-10
Shut down a database instance using SQL*Plus	3-10
Automate Oracle Service startup and shutdown	3-11
Oracle process (UNIX systems)	3-12
Introduction to Oracle processes for UNIX systems	3-12
Manually start Oracle processes	3-13
Start the Oracle listener process	3-13
Initialize all flagged database instances using dbstart	3-14
Initialize a database instance using SQL*Plus utility	3-14
Manually stop Oracle processes	3-15
Stop the Oracle listener	3-15
Shut down all flagged database instances using dbshut	3-16
Shut down a database instance using SQL*Plus	3-16
Automate Oracle processes startup and shutdown	3-17
Automate Oracle processes startup	3-17
Automate Oracle processes shutdown	3-18
Database security (Windows systems)	3-19
Database management	3-19
Database management (Windows systems)	3-19

Database management (UNIX systems)	3-20
Database deletion (Windows systems)	3-21
Semaphores (UNIX) systems	3-22
Introduction to UNIX semaphores	3-22
Oracle use of semaphores	3-22
Common semaphore problems	3-23
NLS_LANG environment variable	3-24
Oracle initialization parameter files	3-25
Oracle online documentation	3-27
Oracle Net implementation	3-27
Oracle Net features	3-27
Configuration files	3-28
Oracle Net assistant	3-29
Service resolution on Teamcenter clients	3-29
Maintaining the Microsoft SQL server database	3-29
Tune the Microsoft SQL Server database	3-29
Start the SQL Server service	3-30
Shut down the SQL Server service	3-30
Database security	3-30
Database deletion	3-31
Error logs	3-31
Implementing Microsoft SQL Net	3-31
ODBC Communication	3-31
Default Net-Library settings	3-31

Chapter

3 *Maintaining the database server*

Maintaining the IBM DB2 server

Tune the IBM DB2 server

For best performance, tune IBM DB2 database settings and services for your site. Tuning any database management system is an iterative maintenance process that requires record keeping and patience to measure, reconfigure, and measure again to optimize performance.

For more information about tuning and maintaining the DB2 server, see the IBM DB2 Information Center:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>

Start the DB2 service

Windows systems:

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 each service name that begins with **DB2-instance-name** and click **Start**.
5. To verify that the DB2 Server service is running, check the **Status** column. When the service is running, the status is **Started**.

Note Alternatively, you can start the DB2 server by browsing to the **DB2-home\sqllib\bin** directory and double-clicking the **db2start** program icon. (Replace **DB2-home** with the home directory of the DB2 installation.)

UNIX or Linux systems:

1. Change to the **DB2-home/sqllib/adm** directory.
Replace **DB2-home** with the home directory of the DB2 instance owner.
2. Enter the following command:

```
db2start
```

Stop the DB2 service

Windows systems:

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 each service name that begins with **DB2–instance-name** and click **Stop**.
5. To verify that the DB2 Server service has stopped, check the **Status** column. When the service not running, the status is blank.

Note Alternatively, you can stop the DB2 server by browsing to the **DB2-home\sqllib\bin** directory and double-clicking the **db2stop** program icon. (Replace **DB2-home** with the home directory of the DB2 installation.)

UNIX or Linux systems:

1. Change to the **DB2-home/sqllib/adm** directory.
Replace **DB2-home** with the home directory of the DB2 instance owner.
2. Make sure the database is not active:

```
db2 list applications for database database-name
```

3. Type the following command:

```
db2stop
```

Open the DB2 Control Center

To open the DB2 Control Center, open a command prompt and enter the following command:

```
db2cc
```

Alternatively, on Windows systems, you can open the DB2 Control Center by clicking the **Start** button and choosing the following menu commands:

Programs→IBM DB2→instance-name→General Administration
Tools→Control Center

For information about using the DB2 Control Center, see the IBM DB2 Information Center:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>

Verify database connectivity

1. Open a DB2 command prompt.

Windows systems:

Start→All Programs→IBM DB2→instance-name→Command Line Tools→Command Window

UNIX or Linux systems:

Open a bash command prompt and change to the DB2 home directory.

2. Enter the database connect command:

```
db2 connect to db-name user db-user using db-pw
```

Replace *db-name* with the database name, *db-user* with the database user name, and *db-pw* with the database user password.

If the database connection is successful, DB2 displays database connection information similar to the following:

```
Database Connection Information
Database server      = DB2/platform DB2-version
SQL authorization ID = db-user
Local database alias = db-name
```

Moving a DB2 database from Windows to Linux

Caution

- This procedure applies *only* to moving a database from Windows to Linux. Do not attempt to move a database from Windows to UNIX using this procedure.
- Make a full backup of the source database.
- Make sure the database server is shut down before you begin this procedure.

1. Export the source database from Windows

- a. Extract the source database schema. This Data Definition Language (DDL) is used to create the source database (schema) on the destination server (if it is not available) using the **db2look** utility:

```
db2look -d src_db -a -l -e -o db2ddl.sql
```

Check the **db2ddl** file for the source database name and replace the same with target database name, if it is different from the source. The DDL script is created in the current working directory.

- b. Export the source database data by entering the following command from your current working directory:

```
db2move source-db export -sn db-user
```

Replace *source-db* with the name of the source database. Replace *db-user* with the database user name. For example:

```
db2move src_db export -sn infodata
```

Run this command from the directory in which you want the export files to be stored. Depending on the size of the database, this process may take significant time to complete. A 15–20 GB database can take more than 30 minutes to export.

- c. Transfer export files to the host on which you want to import the database.

2. Import the database into Linux

- a. On the target database host, verify the following conditions:

- The DB2 server is installed and running.
- The target database is created.
- The current Teamcenter schema exists on the DB2 server.

If schema does not exist, create it by entering the following command from a command line processor or by using a batch file execution, depending on the platform of the target host:

```
db2 -tvf /file-path/db2ddl.sql -z log.txt
```

- b. Import the database.

- To load the data into the target database with large object (LOB) data, enter the following command:

```
db2move target-db load -l /home/userid/lobpath
```

Replace *target-db* with the target database name.

- To load the data on the target database *without* LOB data on Linux, enter the following command from the directory of the load files:

```
db2move target-db load -lo insert
```

Replace *target-db* with the target database name.

- c. Verify the integrity of the migrated database.

Tables in **CHECK PENDING STATE** state are indicated by a **C** in the **status** column. To check which tables are in pending state, type the following command:

```
db2 select tabname from syscat.tables where status='C'
```

The system displays a list of tables that require the **set integrity** statement. To make these tables usable, change the **SET INTEGRITY** value to **IMMEDIATE CHECKED** or **NORMAL** by typing one of the following commands:

```
db2 set integrity for table-name immediate checked
```

```
db2 set integrity for table-name normal
```

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.

For lists of Oracle configuration settings and tuning methods that have the greatest impact on Teamcenter performance, see the *Teamcenter Deployment Guide*, available in the documentation section of Siemens PLM Software's support site. The

Teamcenter Deployment Guide also provides an in-depth review of Oracle database performance issues and diagnosis, and configuration and tuning guidelines.

Note This supplementary maintenance information for Oracle software does not replace the more detailed information in the Oracle product documentation.

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=tceng
```

UNIX or Linux systems:

```
export ORACLE_HOME=ORACLE_HOME
export ORACLE_SID=tceng
```

All examples assume that the database SID is **tceng**.

Manually set the PATH environment variable

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
```

UNIX or Linux systems:

```
export PATH=$PATH:$ORACLE_HOME/bin
```

Manually set the shared library path (UNIX 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:

AIX:

```
export LIBPATH=${LIBPATH}: ${ORACLE_HOME}/lib
```

HP-UX:

```
export SHLIB_PATH=${SHLIB_PATH}: ${ORACLE_HOME}/lib
```

Solaris:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}: ${ORACLE_HOME}/lib
```

It is advantageous for database administrators to define **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 on the Oracle server, it is necessary to start this service because all Teamcenter database requests use the remote connect mechanism.

Process	Description
Database instance	<p>There is one Oracle database service for an Oracle database instance. It must be running for Teamcenter to function properly. The service is:</p> <p style="text-align: center;">OracleServiceSID</p> <p style="text-align: center;"><i>SID</i> represents the database instance system identifier.</p> <p>Starting the instance of an Oracle database is referred to as <i>initializing a database instance</i>. To initialize a database instance, use the Oracle SQL*Plus utility to manually start one database instance defined by the ORACLE_SID environment variable.</p> <p>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.</p>

Use the Oracle SQL*Plus utility to stop a single database instance defined by the **ORACLE_SID** environment variable.

Manually start Oracle services

Before manually starting Oracle Services

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

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**. For more information, see [Installation on Windows Servers Guide](#).

- Control panel startup
 1. Log on to the operating as a user with administrator privilege.
 2. Choose **Start→Settings→Control Panel→Services**.

3. Select the service named **OracleTNSListener**.
4. Click **Start**.
5. Verify that the **OracleTNSListener** service is running by checking the Windows **Services** dialog box for the following entry:

OracleTNSListener **Started** *startup-mode*

startup-mode is either **Automatic** or **Manual**.

6. 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=c:\orant  
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:

OracleTNSListener **Started** *startup-mode*

startup-mode is either **Automatic** or **Manual**.

Start database services

1. Log on to the operating system as a user with administrator privilege.
2. Choose **Start**→**Settings**→**Control Panel**→**Services**.
3. Select the service named **OracleServiceSID** (*SID* is the instance system identifier, for example **tceng**).
4. Click **Start**.
5. 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**.

6. Click **Close**.

*Initialize a database instance using SQL*Plus Utility*

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=D:\oracle\ora92
set ORACLE_SID=tceng
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

The system displays a message similar to the following:

```
:
Oracle Enterprise Edition Release oracle-version - Production
With the Partitioning option
(c) JServer Release oracle-version - Production
```

4. 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
```

The following system message 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

Stop the Oracle listener

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**. For more information about the **listener.ora** file, see [Configuration files](#).

- Control panel shutdown
 1. Log on to the operating system as a user with administrator privilege.
 2. Choose **Start→Settings→Control Panel→Services**.
 3. Select the service named **OracleTNSListener**.

4. Click **Stop**.
5. Verify that the **OracleTNSListener** service has stopped running by checking the Windows **Services** dialog box for the following entry:

OracleTNSListener *startup-mode*

startup-mode is either **Automatic** or **Manual**.

6. 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 the **ORA_DBA** Windows local group.
 2. Manually set the Oracle environment:

```
set ORACLE_HOME=c:\orant  
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:

OracleTNSListener *startup-mode*

startup-mode is either **Automatic** or **Manual**.

Stop database services

1. Log on to the operating system as a user with administrator privilege.
2. Choose **Start**→**Settings**→**Control Panel**→**Services**.
3. Select the service named **OracleServiceSID** (*SID* is the instance system identifier, for example **tceng**).
4. Click **Stop**.
5. 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**.

6. Click **Close**.

*Shut down a database instance using SQL*Plus*

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=D:\oracle\ora92
set ORACLE_SID=tceng
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

4. The system displays a message similar to the following:

```
:
Oracle Enterprise Edition Release oracle-version - Production
With the Partitioning option
JServer Release oracle-version - Production
```

5. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

6. At the **SQLPLUS** prompt, type the following command:

```
shutdown
```

The following system message is displayed:

```
Database closed.
Database dismounted.
ORACLE instance shut down.
```

7. 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

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→Settings→Control Panel→Services**.
3. Select the desired service, for example, **OracleTNSListener** for the Oracle listener service or **OracleServiceSID** (*SID* is the instance system identifier, for example **tceng**).
4. 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.

5. If the startup mode is not **Automatic**, click **Startup** in the **Services** dialog box, change the **Startup Type** to **Automatic**, and click **OK**.
6. Click **Close**.

Oracle process (UNIX systems)

Introduction to Oracle processes for UNIX 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: <ul style="list-style-type: none">• 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 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

Process	Description
	Log writer process; writes the Oracle redo log buffer to a redo log file on disk.

SID 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, **tceng**), **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

Start the Oracle listener process

This example shows the startup of a listener process called **LISENER**. 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=/u01/app/oracle/product/oracle-version  
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 database instances using dbstart

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=/u01/app/oracle/product/oracle-version  
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_tceng  
oracle 1831 1 80 date time ora_pmon_tceng  
oracle 1832 1 80 date time ora_lgwr_tceng  
oracle 1833 1 80 date time ora_smn_tceng  
oracle 1832 1 80 date time ora_reco_tceng  
oracle 1833 1 80 date time ora_ckpt_tceng
```

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 **tceng**.

*Initialize a database instance using SQL*Plus utility*

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=/u01/app/oracle/product/oracle-version  
export ORACLE_SID=tceng
```

3. Set the shared library path environment variable to include the Oracle **lib** directory:

AIX:

```
export LIBPATH=${LIBPATH}: ${ORACLE_HOME}/lib
```

HP-UX:

```
export SHLIB_PATH=${SHLIB_PATH}: ${ORACLE_HOME}/lib
```

Solaris:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}: ${ORACLE_HOME}/lib
```

4. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

The system displays a message similar to the following:

```
:  
Oraclelever Enterprise Edition Release oracle-version - Production  
With the Partitioning option  
(c) JServer Release oracle-version - Production
```

5. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

6. 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.
```

7. The Oracle database is initialized. Exit the Oracle SQL*Plus utility by entering the following at the **SQLPLUS** prompt:

```
exit
```

Manually stop Oracle processes

Stop the Oracle listener

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=/u01/app/oracle/product/oracle-version  
export PATH=$PATH:$ORACLE_HOME/bin
```

The commands assume that Oracle is installed in the **u01/app/oracle/product/oracle-version**; directory. Your location may be different.

3. Stop **tnslnsr** by typing the following command:

```
$ORACLE_HOME/bin/lsnrctl stop listener
```

4. Verify that the **tnslnsr** process is no longer running by entering the following command:

```
ps -ef | grep -v grep | grep tnslnsr
```

There should be no output returned by this command.

Shut down all flagged database instances using dbshut

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=/u01/app/oracle/product/oracle-version  
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 a database instance using SQL*Plus*

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=/u01/app/oracle/product/oracle-version  
export ORACLE_SID=tceng
```

3. Set the shared library path environment variable to include the Oracle **lib** directory:

AIX:

```
export LIBPATH=${LIBPATH}:$ORACLE_HOME/lib
```

HP-UX:

```
export SHLIB_PATH=${SHLIB_PATH}:$ORACLE_HOME/lib
```

Solaris:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$ORACLE_HOME/lib
```

4. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

5. The system displays a message similar to the following:

```
:  
Oracle® Enterprise Edition Release oracle-version - Production  
JServer Release oracle-version - Production
```

6. At the **SQLPLUS** prompt, type the following command:

```
connect / as sysdba
```

The following system message is displayed:

```
Connected.
```

7. At the **SQLPLUS** prompt, type the following command:

```
shutdown immediate
```

The following system message is displayed:

```
Database closed.  
Database dismounted.  
ORACLE instance shut down.
```

8. The Oracle database is shut down. Exit the SQL*Plus utility by typing the following command at the **SQLPLUS** prompt:

```
exit
```

Automate Oracle processes startup and shutdown

Automate Oracle processes startup

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 PLM Software.

Because the Siemens PLM Software-provided sample scripts are highly platform-specific, sample startup scripts, **oracle.daemon**, have been included. The following is a sample startup script provided and the system directories where it must be located on each supported platform:

AIX:

/etc/oracle.daemon

HP-UX:

/sbin/init.d/oracle.daemon

Solaris:

/etc/init.d/oracle.daemon

In addition to the **oracle.daemon** script, a numbered symbolic link to this file must be created in the default run control directory of SVR4 platforms:

AIX:

Edit the **/etc/rc** script by adding the following line to the local customization section to run the **oracle.daemon** script.

```
/etc/oracle.daemon
```

HP-UX:

/sbin/rc2.d/S99oracle.daemon

Solaris:

/etc/rc2.d/S99oracle.daemon

Automate Oracle processes shutdown

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 PLM Software

Because the sample scripts are highly platform-specific, sample shutdown scripts, **oracle.daemon**, have been included. The following is a sample shutdown script provided and the system directories where it must be located on each supported platform:

HP-UX:

/sbin/init.d/oracle.daemon

Solaris:

/etc/init.d/oracle.daemon

In addition to the **oracle.daemon** script, a numbered symbolic link to this file must be created in the default run control directory of SVR4 platforms:

HP-UX:

/sbin/rc2.d/K10oracle.daemon

Solaris:

/etc/rc0.d/K10oracle.daemon

Note There is no available procedure for automating shutdown of Oracle processes on AIX systems.

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\network\admin\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.

For more information, see the *Installation on Windows Servers Guide* and the *Installation on UNIX and Linux Servers Guide*.

Database management

Database management (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=D:\oracle\ora92
set ORACLE_SID=tceng
set PATH=%PATH%;%ORACLE_HOME%\bin
```

3. Start the Oracle SQL*Plus utility:

```
%ORACLE_HOME%\bin\sqlplus
```

The system displays a message similar to the following:

```
:
Oracle Enterprise Edition Release oracle-version - Production
With the Partitioning option
JServer Release oracle-version - Production
```

4. 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
```

Database management (UNIX 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=/u01/app/oracle/product/oracle-version  
export ORACLE_SID=tceng
```

3. Set the shared library path environment variable to include the Oracle **lib** directory:

AIX:

```
export LIBPATH=${LIBPATH}: ${ORACLE_HOME}/lib
```

HP-UX:

```
export SHLIB_PATH=${SHLIB_PATH}: ${ORACLE_HOME}/lib
```

Solaris:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}: ${ORACLE_HOME}/lib
```

4. Start the Oracle SQL*Plus utility:

```
$ORACLE_HOME/bin/sqlplus
```

The system displays a message similar to the following:

```
:  
Oraclelever Enterprise Edition Release oracle-version - Production
```

Jserver Release oracle-version - Production

5. 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.

6. The following system message is displayed:

```
Connected.
```

7. 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';
```

8. To exit the SQL*Plus utility, type the following command from the **SQLPLUS** prompt:

```
exit
```

Database deletion (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→Programs→Oracle→home-name→Database Administration→Database Configuration Assistant**.
3. Select **Delete a database** and click **Next**.
4. Select the service to be deleted (for example, **OracleServiceTC**) 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**.

Semaphores (UNIX) systems

Introduction to UNIX semaphores

UNIX 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.

UNIX 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.

UNIX typically uses many semaphores and allocates them to the system in sets. When the UNIX kernel is configured, the following semaphore parameters are rigidly set and cannot be changed without rebuilding the UNIX kernel and restarting the system:

- Maximum number of semaphores (**SEMMNS**)

The UNIX 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:

```
SEMMNS=[(A=100) + (B=100) + [(C=200) * 2] + [(# of instances=3) * 10] = 630
```

For detailed information about allocating semaphores and rebuilding the UNIX kernel, see the operating system documentation.

- Maximum number of semaphores per set (**SEMMSL**)

The UNIX 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 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 listed in step 46-2.
4. Repeat step 3 until all semaphores owned by Oracle are reallocated.

Common semaphore problems

Oracle problems and errors involving UNIX semaphores often indicate insufficient or improperly configured semaphore resources on that UNIX system. Semaphore resources may need to be optimized before Oracle runs properly.

For detailed information about allocating semaphores and rebuilding the UNIX kernel, see the *Installation on UNIX and Linux Servers Guide*.

- 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 too many semaphores or semaphore sets are already allocated.

The corrective actions for all three of the preceding errors are the same:

- o Check semaphores in use. Verify all unused semaphores are reallocated.
- o Rebuild UNIX 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:

- o ORA-7264 spwat: semop error, unable to decrement semaphore
- o 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

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 UNIX and 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

UNIX and 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

UNIX and 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
```

UNIX and 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 **tnlsnr** (UNIX) or the Windows **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 UNIX:

```
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 **infodba** user as an example, the resulting connect string within Oracle would be:

```
infodba/infodba@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.

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 (UNIX) 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 PLM 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 UNIX is as follows:

HOME/.file-name.ora

TNS_ADMIN/file-name.ora

/etc/file-name.ora (HP-UX and AIX) or ***/var/opt/oracle/file-name.ora*** (Solaris)

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 UNIX, ***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 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 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 over to the appropriate nodes by hand (via **ftp** usually).

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 **TNS_ADMIN** to **%TC_DATA%** in the **tc_profilevars.bat** file, to force Teamcenter to use this file.
- On UNIX, Teamcenter accomplishes this by creating a copy of the **tnsnames.ora** file in the Teamcenter data directory during installation and setting the value of **TNS_ADMIN** 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.

The setting of **TNS_ADMIN** may be overridden from the external environment or commented out in **tc_profilevars**, 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.

For lists of Microsoft SQL Server 2000 or 2005 configuration settings and tuning methods that have the greatest impact on Teamcenter performance, see the *Teamcenter Deployment Guide*, available in the documentation section of Siemens PLM Software's support site. The *Teamcenter Deployment Guide* also provides an in-depth review of Microsoft SQL database performance issues and diagnosis, and configuration and tuning guidelines.

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, double-click **Services**.
4. From the list of services, select **MS SQLSERVER** 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 is not running, the status is blank.

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 2000 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 2000 user account can use trusted connections in either Windows authentication mode or mixed mode.

Siemens PLM 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.

Database deletion

If a Teamcenter database is corrupted beyond repair, the simplest solution may be to delete all information from the database and start again.

To delete a database:

1. Start SQL Server Enterprise Manager and expand **Microsoft SQL Servers**→**SQL Server group**→**Local Server**.
2. Select and right-click **Database** menu under **LOCAL**.
3. Select the corrupted database and delete it.

Error logs

You can view the SQL Server error log using SQL Server Enterprise Manager or any text editor. The most current error log is named **Errorlog** (with no extension) and is located in the **Program Files\Microsoft SQL Server\MS SQL\Log** folder by default.

To view the SQL Server error log:

1. Expand a server group and then expand a server.
2. Expand **Management** and then expand **SQL Server Logs**.
3. Click the SQL Server log.

Error log information is displayed in the details pane.

Implementing Microsoft SQL Net

ODBC Communication

Teamcenter communicates with the SQL Server through the open database connectivity (ODBC) connection. ODBC is an application programming interface (API) that allows a programmer to abstract a program from a database. When programming to interact with ODBC, you need only use the ODBC language (a combination of ODBC API function calls and the SQL language). The ODBC Manager determines how to connect with the type of database the user is targeting. Regardless of the database type used, all calls are to the ODBC API. An ODBC driver must be installed that is specific to the type of targeted database.

Default Net-Library settings

Before running SQL setup, you must enable TCP/IP networking. The setup program installs all Net-Libraries. Following are the default server and client Net-Library settings:

- Server: TCP/IP sockets, shared memory, named pipes.
- Client: TCP/IP sockets, named pipes.

Chapter

4 Teamcenter licenses

Named user licensing	4-1
Common licensing server	4-2
Managing licenses	4-2
Using the convert_license_log utility	4-3
Generating license usage and module usage reports	4-4
Using the LicenseUsedAuditTool tool	4-6

Chapter

4 Teamcenter licenses

Named user licensing

Teamcenter employs *named user licensing*, which ties each user in the system to an available license and ensures the total number of active licenses of each type in the system is always less than or equal to the number of licenses purchased. There can be multiple levels of Teamcenter user licenses to support different roles and uses within an organization. The available license levels and their associated descriptions are available in your license agreement documentation.

Your site's named user licensing requirements are managed with three licensing programs:

- Optional module

Includes all licensed Teamcenter features excluding those associated with seat levels.

Optional module licenses are required to use certain Teamcenter solutions, such as Change Management and Systems Engineering.

- Seat-level licensing

Specifies the base license level associated with each user, such as author, consumer, and occasional user.

This association is mandatory for the users to log on to Teamcenter and use the basic features of the Foundation template (core Teamcenter).

- License bundles

Includes specified Teamcenter features comprising a seat level and/or optional modules. License bundles are then assigned to specific users. 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 when 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.

You can assign license levels to users two ways:

- Use the **Users** pane within the Organization application.

To set a user's license level within Organization, open the **Users** pane and use the **Licensing Level** buttons to set the license level of the user to the desired license level, or use the **License Bundle** box to assign the license bundle. By

default, when creating new users, Teamcenter sets a license value that enables authoring.

- Use the **-licenselevel** or **-licensebundle** arguments with the **make_user** utility to set a user's license level or to assign the license bundle.

For more information about using the **make_user** utility, see the [Utilities Reference](#).

The list of license keys in your site's license file determines which license levels appear. Possible license levels are described in your license agreement documentation. The **license_level** attribute on the **POM_user** class tracks license levels.

Note During upgrade from Teamcenter 2007.1 MP2 or earlier, Teamcenter assigns existing users the lowest available license level by default. After upgrade, Teamcenter administrators must change users to other licensing levels where appropriate.

Common licensing server

The Siemens PLM Software Common Licensing Server daemon, **ugslmd**, enforces license usage. If you maintain installations of Teamcenter 10.1 and Teamcenter 2007.1 MP2 or earlier, you must maintain the old and new license daemons: **ugslmd** for Teamcenter 10.1 and **uglmd** for versions of Teamcenter earlier than Teamcenter 2007.1 MP3.

For information about installing the Siemens PLM Software Common Licensing Server daemon, see the appropriate server installation guide (for [Windows](#) or [UNIX/Linux](#)) or the [Upgrade Guide](#).

For more information about the Siemens PLM Software Common Licensing Server, see the *UGS Licensing User Guide* in the **additional_documentation** directory on the Teamcenter documentation distribution image.

Managing licenses

The list of license keys in your site's license file determines which license levels display. Possible license levels are described in your license agreement documentation.

Make sure you do not exceed your allowed number of licenses. If necessary, you can mark users as inactive using the **-status** argument in the **make_user** utility (or through the **Users** pane within Organization) to reduce the number of licenses being used.

To set an alert when available licenses are low, you can set the **license_warning_level** preference to define the threshold of available Teamcenter feature licenses. Teamcenter displays a warning message when you create or modify a user if the number either equals or falls below the value in the **license_warning_level** preference. The default value is **5**.

In addition to the **license_warning_level** preference, you can set other preferences to configure license management:

- **LicenseUsage_admin_notifier_list**

Specifies the identifiers of the administrative users who receive Teamcenter notifications when the occasional user exceeds the allotted usage and/or grace period limits specified for the occasional user license level.

- **LicenseUsage_days_warning_level**

Specifies the remaining duration in days in the allotted usage when occasional logged-on users start receiving warning messages as a notification as they approach their allotted usage days.

- **LicenseUsage_hours_warning_level**

Specifies the remaining duration in hours in the allotted usage when occasional logged-on users start receiving warning messages as a notification as they approach their allotted usage hours.

- **LicenseUsage_module_usage_warning_level**

Specifies the logins remaining in the month when administrators start receiving warning messages that the module usage is approaching the allowed limit.

- **LicenseUsage_show_userId_in_report**

Activates the display of the actual user ID in the license usage report and the module usage report.

- **Siemens_PL_email_id**

Specifies the valid e-mail ID of the Siemens PLM Software licensing account team that receives notifications when any module usage limit exceeds its allowed usage.

Perform the following tasks to track license usage:

- Track and analyze license usage at your site by using the **convert_license_log** utility to import license log files into Microsoft Excel.

For more information, see [Using the convert_license_log utility](#).

- Generate license usage reports from license logging information stored in the Teamcenter database.

For more information, see [Generating license usage and module usage reports](#).

- Derive license usage reports using the **LicenseUsageAuditTool** tool to process raw FlexNet log files and output summary usage reports in a comma-separated file format (.csv file) that can be opened in Microsoft Excel for further analysis.

For more information, see [Using the LicenseUsedAuditTool tool](#).

Using the **convert_license_log** utility

To make sure you do not exceed your allowed number of licenses, track and analyze license usage at your site using the **convert_license_log** utility to import license log files into Microsoft Excel.

License log files are stored in the **syslog** file. The raw license file includes time stamp, license daemon, license checkin/checkout, feature key, and user ID. For example:

```
6:02:45 (lmgrd) TIMESTAMP 5/20/2007
6:02:47 (ugslmd) OUT: "tol_cavity_milling" smeyer@svli6020
6:04:44 (ugslmd) IN: "tol_cavity_milling" smeyer@svli6020
6:04:51 (ugslmd) OUT: "tol_cavity_milling" smeyer@svli6020
6:11:09 (ugslmd) IN: "tol_cavity_milling" smeyer@svli6020
6:11:10 (ugslmd) OUT: "tol_cavity_milling" smeyer@svli6020
6:11:20 (ugslmd) IN: "tol_cavity_milling" smeyer@svli6020
12:02:45 (lmgrd) TIMESTAMP 5/20/2007
12:53:51 (ugslmd) OUT: "gateway" jdahlke@ahi6w022
12:54:02 (ugslmd) OUT: "ufunc_execute" jdahlke@ahi6w022
12:54:02 (ugslmd) IN: "ufunc_execute" jdahlke@ahi6w022
12:58:25 (ugslmd) OUT: "cam_base" jdahlke@ahi6w022
12:58:42 (ugslmd) OUT:
```

The license log file can be output as a text file or Microsoft Excel file. For example:

A927	Date	Time	License Server	Operation	Feature Key	User and Server
1	923	5/19/2010	14:24:54 (ugslmd)	IN:	cavity_milling	jdahlke@svli6020
	924	5/19/2010	14:26:54 (ugslmd)	IN:	cam_base	jdahlke@svli6020
	925	5/19/2010	14:26:54 (ugslmd)	IN:	gateway	jdahlke@svli6020
	926	5/19/2010	22:45:48 (ugslmd)	OUT:	visview_mockup	gerth@ahi3p053
	927	5/19/2010	22:45:49 (ugslmd)	OUT:	visview_publish_write	gerth@ahi3p053
	928	5/19/2010	22:45:49 (ugslmd)	IN:	visview_publish_write	gerth@ahi3p053
	929	5/19/2010	22:46:03 (ugslmd)	OUT:	visview_animation_cre	gerth@ahi3p053
	930	5/19/2010	22:46:04 (ugslmd)	IN:	visview_animation_cre	gerth@ahi3p053
	931	5/19/2010	22:46:05 (ugslmd)	OUT:	visview_animation_cre	gerth@ahi3p053
	932	5/19/2010	22:46:05 (ugslmd)	IN:	visview_animation_cre	gerth@ahi3p053
	933	5/19/2010	22:46:06 (ugslmd)	OUT:	visview_clearance_mar	gerth@ahi3p053
	934	5/19/2010	22:46:07 (ugslmd)	IN:	visview_clearance_mar	gerth@ahi3p053
	935	5/19/2010	22:46:07 (ugslmd)	OUT:	visview_clearance_mar	gerth@ahi3p053
	936	5/19/2010	22:46:07 (ugslmd)	IN:	visview_clearance_mar	gerth@ahi3p053
	937	5/19/2010	22:46:08 (ugslmd)	OUT:	visview_vsa_buildmod	gerth@ahi3p053
	938	5/19/2010	22:46:09 (ugslmd)	IN:	visview_vsa_buildmod	gerth@ahi3p053
	939	5/19/2010	23:17:26 (ugslmd)	IN:	visview_mockup	gerth@ahi3p053

For more information about this utility, see the [Utilities Reference](#).

Generating license usage and module usage reports

License usage details are logged in the Teamcenter database. The availability of license logging within the Teamcenter database provides an alternative to using FlexNet (monitoring the FlexNet log and running the **LicenseUsageAuditTool** tool to generate useful output) or other tools such as the **Imstat** utility or custom reporting tools. Additionally, accessing license logs from the Teamcenter database provides a secure method of hiding user identity in the output reports by using hash table encoding to generate pseudo user IDs.

The following license usage details are logged in the Teamcenter database. (Previous to Teamcenter 9.0, only the last logon time and date was logged.)

- Listing of users, seat, and feature usage per month.
- Hours of usage 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.
- Supports 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 usage per month.
- Number of logons per month.
- Any usage violations.

You can generate reports of this information using the **License Usage Report** and **Module Usage Report** from the Report Builder reports:

1. In the rich client, choose **Tools® Reports® Report Builder Reports**.
The **Report Generation Wizard** appears.
2. At the **Select Report** step, select one of the following reports.
 - **License Usage Report**
Provides a summary of each named user's usage per month of seat-level licenses (for example, for authors and consumers).
 - **Module Usage Report**
Provides a summary of usage per month of optional module licenses (for example, change author, requirements access, and so on).
3. Click **Next**.
4. At the **Fill in Criteria** step, specify the desired criteria for the report.
5. From the **Report Stylesheets** list, select the **admin_license_usage_text.xls** style sheet for the **License Usage Report**, or the **module_license_usage_text.xls** style sheet for the **Module Usage Report**.
6. Click **Finish**.

The report is generated and displayed in comma-separated format. The output can be saved as a **.csv** file and opened in Microsoft Excel.

The output is similar to the FlexNet-based report. However, the pseudo IDs are encrypted more securely, and the feature keys are grouped together on one line. For example, following is output of the **License Usage Report**.

user id	pseudo id	seat level	license bundle	feature keys	month	used hours
user5	5a39bead318f3069;teamcenter_consumer			teamcenter_consumer	13-Mar	1
user7	cd8b0429ce02c712;teamcenter_author			teamcenter_author	13-Mar	265
user1	0a041b9462caa4a3	teamcenter_author	STUDENT	teamcenter_author	13-Mar	1

License Usage Report output

The pseudo ID always displays in the report. The **LicenseUsage_show_userId_in_report** preference determines whether the actual user ID displays as well. By default, this preference is set to **true**, displaying the actual user ID along with the pseudo ID.

Using the LicenseUsedAuditTool tool

The **LicenseUsageAuditTool** tool augments the reports available from the [convert_license_log](#) utility.

The tool analyzes the raw FlexNet log file and outputs a **.csv** file with the following information for the period covered by the log file:

- Listing of users, seat, and feature usage per month.
- Hours of usage 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.
- Days of usage per month.
- Number of logons per month.
- Any usage violations.

For example:

Pseudo User	Seat	Feature	Bundle	SeatUsage	Month	Hours	Days	Logins	Usage Violation	U
User0000002	visview_mockup	visview_mockup		Yes	Oct-12	5	1	4	No	
User0000002	visview_mockup	visview_publish_write			Oct-12	4	1	3	No	
User0000002	visview_mockup	visview_vispublish			Oct-12	4	1	2	No	

This tool is stored in the **additional_applications\LicenseUsageAuditTool** directory, in the **LicenseUsageAuditTool.zip** file. The directory contains the following elements:

- In the **\bin** directory, the **LicenseUsageAuditTool.bat** and **LicenseUsageAuditTool.sh** files.
The Windows (**.bat**) and UNIX (**.sh**) scripts to run the tool.
- In the **\conf** directory, the **AuditUsage.xml** file.
The input XML configuration file.
- In the **\lib** directory, the **lib\LicenseUsageAuditTool.jar** file.
The JAR file containing compiled Java classes for use with the tool.

To use this tool:

1. Set the **JAVA_HOME** environment variable to the Java home directory.
2. Set the tool's arguments as follows:
 - **-i**
Specify the full path and file name of the input file. It must be a FlexNet log file.
 - **-o**
Specify the output file name. It must be a **.csv** file.

- (Optional) **-pseudouser**

Use this argument to output user names as pseudo user names. Use this option to circumvent attributing individual usage to actual users, for example, to protect the identity of individual user usage where required by labor laws.

For example, run the following command on one line:

```
LicenseUsageAuditTool.bat  
-pseudouser -i D:\logs\ugslicensing.log -o D:\logs\output.csv
```

Chapter

5 *Teamcenter client communication system (TCCS)*

Managing Teamcenter client communication system (TCCS)	5-1
TCCS configuration files	5-3
Orientation to TCCS configuration files	5-3
tccs.xml file	5-3
fwdproxy_cfg.properties file	5-6
reverseproxy_config.xml file	5-8
Configuring multiple TCCS environments	5-8
TCCS and container applications	5-9
Administering TCCS and its container applications	5-9
Administering the Teamcenter server proxy	5-10
Administering the FCC	5-12
Administering TcMEM	5-12
Administering proxy support for clients not integrated with TCCS	5-12
TCCS logging	5-13

Chapter

5 *Teamcenter client communication system (TCCS)*

Managing Teamcenter client communication system (TCCS)

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.

For more information, see [Administering the Teamcenter server proxy](#).

- 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.

For more information, see [Administering 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.

For more information, see [Administering 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 provides the following features to supported clients:

- Forward proxy support

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

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.

TCCS installation is included in a Teamcenter client installation.

For more information about client installations, see the [Installation on Windows Clients Guide](#), [Installation on Linux Clients Guide](#), or [Installation on Macintosh Clients Guide](#).

Clients using TCCS can use client-certificate authentication through secure sockets layer (SSL) settings.

For more information about client-certificate authentication, see the [Security Services Installation / Customization](#) guide and the [Installation on Windows Clients Guide](#) and the [Installation on Linux Clients Guide](#).

TCCS configuration files

Orientation to TCCS configuration files

Configure TCCS using the following configuration files:

- **tccs.xml**

For more information, see [*tccs.xml file*](#).

- **fwdproxy_cfg.properties**

For more information, see [*fwdproxy_cfg.properties file*](#).

- **reverseproxy_config.xml**

For more information, see [*reverseproxy_config.xml file*](#).

These files are stored in a separate location from the application. The default location for these files is a platform-specific path.

- On Windows systems, the application first checks the `%USERPROFILE%\Siemens\cfg\tccs\%TCCS_CONFIG%` path for the files. If the files are not found, the application checks the `%ALLUSERSPROFILE%\Siemens\cfg\tccs\%TCCS_CONFIG%` path.
- On UNIX and Linux systems, the application first checks the `$HOME/Siemens/cfg/tccs/$TCCS_CONFIG%` path for the files. If the files are not found, the application checks the `etc/Siemens/cfg/tccs/%TCCS_CONFIG%` path.

The **TCCS_CONFIG** environment variable is defined during the installation process. By default, this is set to **Teamcenter**.

You can define a custom location using the **TCCS_CONFIG_HOME** and **TCCS_CONFIG** environment variables. The custom location is formed on Windows systems as `%TCCS_CONFIG_HOME%\%TCCS_CONFIG%` and on UNIX/Linux systems as `$TCCS_CONFIG_HOME/$TCCS_CONFIG`.

If the application cannot find the configuration files, TCCS, FCC, and TcMEM use default configuration values.

tccs.xml file

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.

- o **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.

- o **allowuntrustedcertificates**

Allows TCCS to accept server certificates that are not signed by a trusted certificate authority.

- o **connectiontimeout**

Sets the time period that a connection remains idle before it is closed.

- o **httpversion**

Indicates the HTTP protocol version. The default is version **1.1**.

Note HTTP version 1.1 supports chunking; version 1.0 does not.

- o **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.

- o **keystorepassword**

Specifies the password for the Java keystore. The value is set into the Java system's **javax.net.ssl.keyStorePassword** property.

- o **maxconnectionsperhost**

Sets the maximum number of connections to the proxy server from a single host. The default value is **8**.

- o **maxretryreverseproxy**

Set the maximum number of times an HTTP request is attempted when receiving an authentication challenge from a reverse proxy server.

- o **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.

- o **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.

- o **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.

- o **totalmaxconnections**

Sets the maximum number of connections to the proxy server from all hosts.

- o **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.

- o **truststorepassword**

Specifies the password for the Java trust store. The value is set into the Java system's **javax.net.ssl.trustStorePassword** property.

- o **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.

For more information about configuring a Kerberos authentication protocol, see [*Security Services Installation / Customization*](#).

- o **kerberosconfig**

Set **enable** to **true** to configure support for Kerberos authentication in TCCS. The default value is **false**.

- o **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**.

- o **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:

<http://docs.oracle.com/javase/6/docs/technotes/guides/security/jgss/tutorials/KerberosReq.html>

fwdproxy_cfg.properties file

The **fwdproxy_cfg.properties** file contains forward proxy related configuration properties used by TCCS for server requests. All of these values are set in the Teamcenter Environment Manager (TEM) **TcCS Configuration Selection** and **TcCS Settings** panels. The file contains the following properties:

Note

Whether your network uses IPv6 (128-bit) or IPv4 (32-bit) addresses, use host names in URLs wherever possible so the domain name service (DNS) can determine which IP address should be used.

If you must use IP addresses and your network uses IPv6 addresses, enclose the literal IPv6 address in square brackets, for example:

http://[2001:db8:ffff:1:101:12ff:de13:1322]:9043/tc

- **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:

- o **browser**

Uses the Web browser proxy settings.

If you have more than one browser installed, your designated default browser is used.

- o **network**

Detects the proxy settings from the network the client is on.

- o **url**
Retrieves the proxy autoconfiguration (PAC) file from the URL set in the **tcpProxy.connection.url** property.
- o **manual**
Uses the proxy settings set in the manual configuration properties. These are the properties prefixed with **tcpProxy.connection.manual** in this file. The manual configuration properties are ignored if the **tcpProxy.connection.type** attribute is not set to **manual**.
 - **tcpProxy.connection.url**
Sets the URL the TCCS application uses to retrieve the PAC file. This property must have a value if the **tcpProxy.connection.type** attribute is set to **url**.
 - **tcpProxy.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:
 - o **tcpProxy.connection.manual.all.port**
Sets the port number for the proxy server for all protocols.
 - o **tcpProxy.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.
 - o **tcpProxy.connection.manual.http.port**
Sets the port number for the proxy server for the HTTP protocol.
 - o **tcpProxy.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.
 - o **tcpProxy.connection.manual.https.port**
Sets the port number for the proxy server for the HTTPS protocol.
 - o **tcpProxy.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.
 - o **tcpProxy.connection.manual.socks.port**
Sets the port number for the proxy server for the SOCKS protocol.
 - o **tcpProxy.connection.manual.exceptions**

Contains a semicolon-delimited list of hosts where direct connections can be attempted.

- o **tcpProxy.advanced.address_caching**

Indicates whether resolved proxy addresses are cached based on their host and port. Valid values are **on** and **off** (case-sensitive)

- o **tcpProxy.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.

Note

To use WebRAID with a forward proxy, the forward proxy host and port for the appropriate protocols (HTTP or HTTPS), must be specified in the **FMS_HOME\fcc.properties** file as follows:

```
http.proxyHost=forward-proxy-host  
http.proxyPort=forward-proxy-server-port  
https.proxyHost=forward-proxy-host  
https.proxyPort=forward-proxy-server-port
```

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. The criteria list of values can be set in TEM.

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>  
<criteria>  
    <header name="server" value="SiteMinder"/>  
    <form action="smloginform"/>  
</criteria>
```

Configuring multiple TCCS environments

A single TCCS configuration can contain multiple environments, providing support for multiple versions or server databases. For example, you might want to install some TCCS environments with Security Services (SS), some environments without SSO, some environments on one server, and other environments on another server.

If multiple environments are configured, all environments display 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. The **id** corresponds to **SSO AppID**.

- **tcserver**

The **tcserver** service endpoint is the URL of the Web tier deployment.

To create multiple environments in TEM:

1. In the **Environment Settings for Client Communication System** panel, click **Add** to add an additional TCCS environment.
2. For each environment, specify a name and the URI.

Note Whether your network uses IPv6 (128-bit) or IPv4 (32-bit) addresses, use host names in URIs wherever possible so the domain name service (DNS) can determine which IP address should be used.

If you must use IP addresses and your network uses IPv6 addresses, enclose the literal IPv6 address in square brackets, for example:

http://[2001:db8:ffff:1:101:12ff:de13:1322]:9043/tc

3. (Optional) Specify a tag to identify the environment.

This is useful if numerous environments are configured, resulting in a long list of environments displaying to the user at logon. To filter this list, use the **Tag** field to create a filtering pattern.

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.

4. Specify the SSO information for the three environments using SSO.
5. In the **Client Tag Filter** panel, specify the filtering pattern you want to apply to the environment list that displays to the user at logon.

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.

TCCS and container applications

Administering TCCS and its container applications

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**.

```
<tccsconfig>
  <!--List of applications configured with TCCS -->
  <tccsapps>
    <tccsappconfig name="TcServerProxy" initialize="true" configfile="tcserverproxy.xml"/>
    <tccsappconfig name="FMSClientCache" initialize="true" configfile="fcc.xml"/>
    <tccsappconfig name="TcModelEventManager" initialize="true" configfile="" />
  </tccsapps>
```

Consider the following behavior when working with the TCCS container:

- **TcProxyClient**

The container initializes the **TcProxyClient** component that is the forward and reverse proxy support library for TCCS.

- **Idle shut down**

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 will 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.

For information about correctly stopping the TCCS container, see [Shutting down a TCCS/FCC instance](#).

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 (UNIX).

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. 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 **tspstat** 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 **tcserverproxy.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**

Set 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**.

- **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

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.
For more information about configuration elements and cache parameters, see *Overview of the FMS client configuration file*.

- 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 PLM Software clients are integrated with TCCS, including:

- Teamcenter Systems Engineering
- Teamcenter thin client

Consider the following when administering proxy support for clients not integrated with TCCS:

- Forward proxy support

A nonintegrated 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 nonintegrated 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.

For more information, see [Security Services Installation / Customization](#).

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\process\
tccs_os-user-name_%TCCS_CONFIG_host-name.log**

- UNIX systems:

**\$HOME\user-name\Siemens\logs\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 **log4j.xml** file.

- **log4j.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:

- o **logger name**

Specifies the name of the MLD logger.

- o **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**.

- o **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.

Chapter

6 *Server manager*

Introduction to the server manager	6-1
Server manager prerequisites	6-2
Server manager properties files	6-3
Configuring property files	6-3
Global pool properties	6-3
Pool-specific configuration tuning	6-5
Pool-specific configuration tuning recommendations	6-5
Setting PROCESS_CREATION_DELAY	6-7
Setting the PROCESS_MAX parameter	6-8
Setting the PROCESS_TARGET parameter	6-8
Setting the PROCESS_WARM parameter	6-9
Monitoring	6-11
Introduction to monitoring	6-11
Server manager monitoring	6-12
Server manager monitoring metrics	6-12
Configure monitoring with the poolMonitorConfig.xml file	6-13
Sample poolMonitorConfig.xml code	6-15
Configure monitoring with the server manager administrative interface	6-17
Teamcenter server monitoring	6-19
Teamcenter server monitoring metrics	6-19
Configure monitoring with the serverMonitorConfig.xml file	6-20
Sample serverMonitorConfig.xml code	6-22
Configure monitoring with the server manager administrative interface	6-23
Monitoring system alerts	6-25
Automatic metric collection	6-26
Automatic log level change	6-27
Server manager logging	6-27
Server manager logging levels	6-27
Configuring server manager logging	6-28
Configure server manager logging in the J2EE server manager administrative interface	6-29
Dynamically changing logging levels of business logic servers	6-29
Configure business logic server manager logging in the J2EE server manager administrative interface	6-29
Changing SQL logging behavior	6-31
Configuring Teamcenter server journaling	6-32

J2EE server manager administrative interface	6-32
Using the J2EE server manager administrative interface	6-32
Start the J2EE administrative interface	6-33
Administering the pool's server manager	6-34
Administering monitoring options for the server manager	6-35
Administering Teamcenter servers	6-35
Administering monitoring options for Teamcenter servers	6-36
Managing the HTML adapter	6-37
Viewing JMX server details	6-37
Configuring logging and journaling for Teamcenter servers	6-38
Configuring logging for the pool's server manager	6-39
Managing CORBA IORs	6-40
View IOR addresses	6-40
.NET server manager administrative interface	6-40
Start the .NET administrative interface	6-40
Global pool configuration	6-41
Global Configuration view	6-41
Setting global pool parameters	6-41
Viewing server manager instances	6-41
Restarting warm servers	6-41
Server manager configuration	6-42
Server manager views	6-42
Viewing server manager status	6-42
Configuring server manager pools	6-43
Viewing Teamcenter server instances	6-43
Performing server manager operations	6-43
Using the .NET server manager administrative interface	6-44
Using third-party applications to view server manager administration data . . .	6-45

Chapter

6 *Server manager*

Introduction to the server manager

The *server manager* (sometimes called a pool manager) is a tool to manage system resources, allowing you to maximize server availability while minimizing resources.

Use the server manager to:

- Manage the server pool, using either of the out-of-the-box server manager administrative interfaces or third-party applications.

For information about using the J2EE server manager administrative interface, see [Administering the pool's server manager](#).

For information about using the .NET server manager administrative interface, see [Global Configuration view](#).

For information about using third-party applications, see [Using third-party applications to view server manager administration data](#).

- Manage individual Teamcenter servers, using either of the server manager administrative interfaces.

For information about using the J2EE server manager administrative interface, see [Administering Teamcenter servers](#).

For information about using the .NET server manager administrative interface, see [Server manager views](#).

- Monitor the Web tier for critical events using either the **webtierMonitorConfig.xml** file or the Web tier administrative interface.

For more information about using the XML file, see [Configure monitoring with the webtierMonitorConfig.xml file](#).

For more information about using the Web tier administrative interface, see [Start the administrative interface](#).

- Monitor the server manager for critical events, using either the **poolMonitorConfig.xml** file or the J2EE server manager administrative interface.

For more information about using the XML file, see [Configure monitoring with the poolMonitorConfig.xml file](#).

For more information about using the J2EE server manager administrative interface, see [Administering monitoring options for the server manager](#).

- Monitor Teamcenter servers for critical events, using either the **serverMonitorConfig.xml** file or the J2EE server manager administrative interface.

For more information about using the XML file, see [*Teamcenter server monitoring metrics*](#).

For more information about using the J2EE server manager administrative interface, see [*Administering monitoring options for Teamcenter servers*](#).

- Manage Teamcenter logging levels.

For more information, see [*Server manager logging levels*](#).

Alternatively, you can use a third-party application to manage all server manager data.

For more information, see [*Using third-party applications to view server manager administration data*](#).

Server manager prerequisites

Before using the server manager, you must:

- Install the server manager.

For more information, see Teamcenter Server Installation in either the [*Installation on UNIX and Linux Servers Guide*](#) or the [*Installation on Windows Servers Guide*](#).

- Deploy the Teamcenter Web tier application (EAR file bundling a WAR file).

For more information, see the [*Web Application Deployment Guide*](#).

- Configure global pool properties.

Configuring server pool time-outs for different server states allows you to maximize server availability.

For more information, see [*Global pool properties*](#).

- Configure pool-specific properties.

Once server pools are configured, each individual machine should have its pool-specific configuration tuned. This is of particular concern if each individual machine differs greatly in the number and power of its CPUs from each other.

For more information, see [*Pool-specific configuration tuning recommendations*](#).

- (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.

For more information, see [*Web tier monitoring*](#).

- Start the server manager.

If you install the **J2EE based Server Manager** or **.NET based Server Manager** features, you must start the appropriate server manager to enable four-tier rich clients to connect to the corporate server.

For information, see the [Installation on UNIX and Linux Servers Guide](#) or the [Installation on Windows Servers Guide](#).

Server manager properties files

Configuring property files

Use the following property files to configure server manager behavior:

- **globalPoolConfig.properties**

Used for setting configurations across all pools. This is an efficient option for making use of parallel CPU and memory resources to run Teamcenter servers.

This file is stored in the deployed Web application EAR file during **insweb** installation.

For more information, see [Global pool properties](#).

- **serverPool_{database-name}.properties**

Used for pool-specific tuning of each individual machine. This is of particular importance if each machine differs greatly in the number and power of its CPUs from each other.

This file is stored in the **TC_ROOT/pool_manager/** directory.

For more information, see [Pool-specific configuration tuning recommendations](#).

- **pref_export.xml**

Used for configuring Web tier metrics and logging behavior.

This file is stored in the **TC_ROOT/pool_manager/** directory.

For more information, see [Web tier monitoring](#).

Global pool properties

Global pool properties are set in the **globalPoolConfig.properties** file and stored in the deployed Web application EAR file during **insweb** installation. Time-out parameters are set globally across all pools. Pool sizing parameters must be configured individually for each pool.

For more information about pool-specific parameters, see [Pool-specific configuration tuning recommendations](#).

Parameters include both hard and soft time-outs.

- Soft time-outs

Apply only when the number of servers in a server pool exceeds the **PROCESS_TARGET** parameter configured for the pool manager.

- Hard time-outs

Always apply, regardless of the status of the server pool.

Time-out parameters are available for the following server modes. The client controls the mode of its assigned server at any given moment.

- Edit mode

The client may switch its server to this mode when it (or a user) is making updates to server data that is not yet committed to the database. If the server is lost, these changes are lost.

For example, Structure Manager uses edit mode to allow users to edit a BOM structure through multiple operations that change temporary data in the server and client until the user saves the data.

- Read mode

The client may switch its server to this mode when the client's requests have set a temporary state in the server to be used by subsequent requests. If the server is lost, the client may require restart, and there may be performance issues as the client becomes consistent with a new server, but no significant user work is lost or corrupted.

For example, the rich client's initial mode is read mode.

- Stateless mode

This is the default mode for a server. The client uses this mode when no requests depend on the state that a previous request has made to the server. If the server is lost, the next request can be executed on a new server without functional issues except for the performance of assigning a new server.

For example, the Web client is stateless.

There is a time-out parameter for each combination of soft and hard time-outs combined with each of the three modes. For example, the **SOFT_TIMEOUT_EDIT** parameter applies to servers in edit mode when the server pool exceeds the value set by the **PROCESS_TARGET** parameter.

There are two additional time-out parameters.

- **USER_TIMEOUT_STATELESS**

Configures the server idle time in seconds. This timeout applies after a user hits the limit defined by the **PROCESS_MAX_PER_USER** value.

- **QUERY_TIMEOUT**

Configures the maximum time a server is allowed to process a single request. If this time is exceeded the server is terminated. A value of **0** turns off the query time-out, allowing a server to continue processing a request indefinitely.

- **ASSIGNMENT_TIMEOUT**

Timeout (in seconds) for a server assignment to be completed. This includes the time for the server to authenticate the user credentials and perform user-specific initialization.

The following excerpt from a **globalPoolConfig.properties** file is provided for illustration purposes only. This file is placed in the deployed Web application EAR

file during **insweb** installation. To override the values, **insweb** can be rerun to update these values in a new EAR. Alternately, a copy from the **insweb** staging area can be placed in the J2EE application server startup directory.

```
CACHE CONFIG PATH=TreeCacheMcast.xml
PROCESS MAX_PER_USER=0
QUERY TIMEOUT=0
SOFT TIMEOUT EDIT=7200
SOFT TIMEOUT READ=3600
SOFT TIMEOUT STATELESS=1200
HARD TIMEOUT EDIT=28800
HARD TIMEOUT READ=28800
HARD TIMEOUT STATELESS=28800
USER TIMEOUT STATELESS=0
ASSIGNMENT_TIMEOUT=180
```

You may want to increase the values of some of the **SOFT_TIMEOUT** values to reduce CPU overhead if these time-outs are common. The time-out configuration values are in seconds. Also, though the edit soft time-out default is 7200 seconds (two hours), the consequences are higher for such a time-out, and it may be desirable to increase that value as well.

Pool-specific configuration tuning

Pool-specific configuration tuning recommendations

Tune the server manager configuration to make the best use of specific system resources to maximize server availability while minimizing resources.

Each individual machine should have its pool-specific configuration tuned. This is of particular concern if each individual machine differs greatly in the number and power of its CPUs from each other.

The pool-specific parameters are set in the **TC_ROOT\pool_manager\serverPooldatabase-name.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.

- **PROCESS_WARM**

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 due to 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.

Additional parameters provide the following information. Typically, the default settings are adequate and do not require additional tuning.

- **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**.

- **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**.

- **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_HEARTBEAT_INTERVAL**

Determines the time (in seconds) between license heartbeat calls sent from the manager to each Teamcenter server. The default setting is **720**.

- **ENABLE_SERVER_HEARTBEAT**

Determines whether server heartbeats are enabled. Server heartbeats are the time (in seconds) between license heartbeat calls sent from the manager to each Teamcenter server. The default setting is **0** (off). Enable heartbeats by setting to any nonzero value.

- **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**.

- **ASSIGNMENT_RETRY_LIMIT**

Determines the number of attempts to update **TreeCache** for a new server assignment before returning an error. The default setting is **3**.

- **ASSIGNMENT_RETRY_WAIT_PERIOD**

Determines the delay (in milliseconds) before retrying a **TreeCache** update for a new server assignment. The default setting is **200**.

- **SERVER_HOST**

Defines the host name (or IP address) on which the Teamcenter server listens for CORBA 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**.

Setting PROCESS_CREATION_DELAY

It is critical that the rate of starting new Teamcenter servers not overwhelm the processing resources available on the pool machine. Therefore, Siemens PLM Software recommends the tuning process start by choosing an optimal setting for **PROCESS_CREATION_DELAY**.

1. For your specific installation configuration and machine, measure the amount of operating system processing resources consumed by a Teamcenter server to be 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 can be expressed as *[warmCPUSec]* in units of CPU seconds.

2. Determine what fraction of the machine can be dedicated to starting up new server in response to logon demand during the peak usage of the system. Note that processing resources used to start new servers is resource that is not available to support established active session transactions. Siemens PLM Software recommends values for this between 0.4 and 0.7. This 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 optimum process creation delay in units of milliseconds (mSec) as follows:

$$[optimumDelay] = ([warmCPUSec] * [1000 mSec / Sec]) / ([warmFraction] * [numCPUs])$$

5. Set **PROCESS_CREATION_DELAY** to an increasing value list of delays, starting with the optimum (or slightly under), and generally ramping up to

about 60000 milliseconds. The pool manager moves across the list based on number of consecutive start failures.

Example:

```
[warmCPUSec] = 8 seconds
[warmFraction] = 0.6
[num_CPUs] = 2
[optimumDelay] = (8 * 1000) / (0.6 * 2) = 6667
PROCESS_CREATION_DELAY = 6667 7000 12000 24000 40000 60000
```

By default, **PROCESS_CREATION_DELAY** is not explicitly placed in the file. The default value for **PROCESS_CREATION_DELAY** is the list **2000 2000 8000 16000 30000 60000** (values in milliseconds), so that if the default were used on the machine, a demand for > 100% CPU would occur whenever the pool manager tried to ramp up more than one server. To override the default, the entire line must be added to the file.

Note **PROCESS_CREATION_DELAY** is the most important parameter to set to prevent CPU overload on pool servers due to starting multiple new servers.

Setting the **PROCESS_MAX** parameter

The only concern for limiting the maximum number of Teamcenter servers on a machine might be memory consumption. The incremental free memory per server can be expected to vary a bit from installation to installation, but is on the order of 30 megabytes.

Determine the maximum amount of RAM on the machine that you want do 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

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.

An example is **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 $(200 - 100) * 5000 = 500,000 \text{ milliseconds} = 500 \text{ seconds} = 8.3 \text{ minutes}$ 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

This setting is the most difficult to optimize, because it requires an estimate of the burst rate of logons that may occur. The following discussion examines some of the theory behind what drives a good value. After the theory, more practical statements about tuning this value are presented.

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.

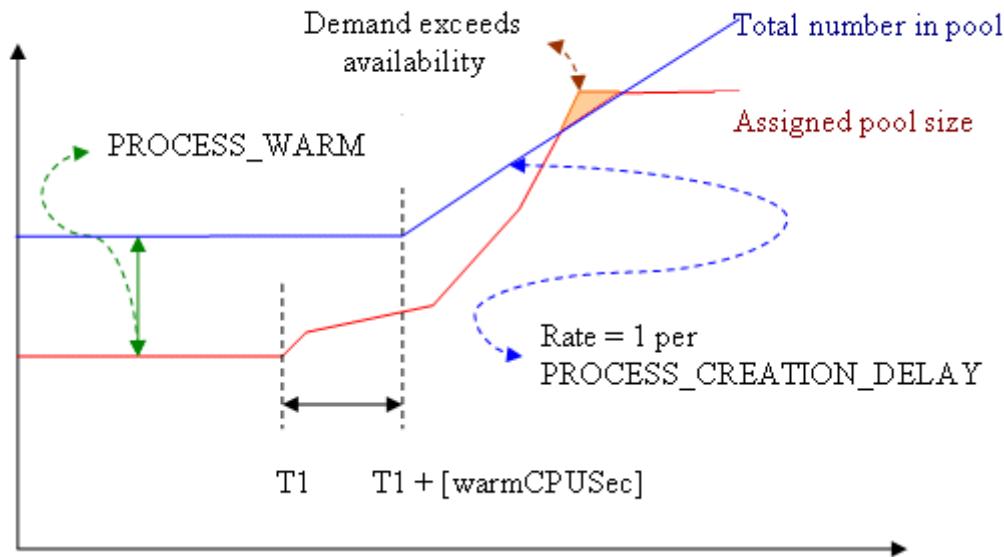
As defined at the start, **PROCESS_WARM** is a desired number of Teamcenter servers to be warm, but not assigned.

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 they are added to the pool at a maximum rate of one every **PROCESS_CREATION_DELAY**.

The following graphic illustrates an assignment burst that starts at time T1, 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 $(1000 \text{ mSec/Sec}) / \text{PROCESS_CREATION_DELAY}$.

- 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 $\text{PROCESS_WARM} + (1000 * ([burst_interval] - [warmCPUSec])) / \text{PROCESS_CREATION_DELAY}$ 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.

Monitoring

Introduction to 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, as well as allowing you to take corrective action.

Different 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.

For information, see [Web tier monitoring](#).

- Server manager

Monitor pool manager events such as login time, server crashes, and various types of server time-outs.

For more information, see [Server manager monitoring metrics](#).

- Teamcenter servers

Monitor server events on specific Teamcenter servers such as out of memory errors, long running queries, and memory use.

For more information, see [Teamcenter server monitoring metrics](#).

- File Management System (FMS) components

Monitor Multi-Site events such as global memory pool events, route events, and ticket events.

For more information, see [Introduction to File Management System monitoring](#).

- Translation components

Monitor translation events such as dispatcher client events, dispatcher scheduler events, and dispatcher module events.

For information about setting debug levels for the dispatcher services, see [Getting Started with Dispatcher \(Translation Management\)](#).

You can configure monitoring behavior for each of these Teamcenter elements from either a J2EE-based administrative interface or from XML configuration files.

Both configuration methods require you to first define the **EmailResponder** element (specifying how and where e-mail 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.

Control the frequency of e-mail notifications and event logging by specifying suppression periods.

Example

You have configured the monitoring of business logic server crashes to send e-mail 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. E-mail 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. E-mail notification is sent. Once again, the suppression clock starts.

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 threshold values slightly higher than normal activity levels.

Server manager monitoring

Server manager monitoring metrics

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive e-mail 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.
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.

Metric	Description
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.

Configure server manager monitoring using either:

- The *TC_ROOT/pool_manager/poolMonitorConfig.xml* file.

For more information about using the XML file, see [Configure monitoring with the poolMonitorConfig.xml file](#).

- The J2EE server manager administrative interface.

For more information about using the J2EE server manager administrative interface, see [Administering monitoring options for the server manager](#).

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 e-mail notifications are generated from the *TC_ROOT/pool_manager/poolMonitorConfigInfo.xml* file. (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 monitoring with the poolMonitorConfig.xml file

1. Open the *TC_ROOT/pool_manager/poolMonitorConfig.xml* file.

2. 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 e-mail 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 e-mail responder. Multiple e-mail responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the e-mail protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the e-mail 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 E-mails are sent.

- **toAddress**

Specify the address to which the notification E-mails are sent.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress e-mail notification of critical events.

For more information, see the suppression period example in [Introduction to monitoring](#).

- **emailFormat**

Specify the format in which the e-mail 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.

For more information, see the suppression period example in [Introduction to monitoring](#).

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 server manager administrative interface) for this metric.

This is the default setting.

- **Alert**

Collect metric data, display results in the MBean view for this metric, and send e-mail notifications when critical events occur.

- **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.

Server manager monitoring is enabled for the metrics you configured.

Sample poolMonitorConfig.xml code

In the following example, two **EmailResponder** elements are configured. The majority of e-mail notifications are sent to **admin1@company.com**, but e-mail 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 e-mail 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"/>
      <hostAddress value="svclsmtp.company.com"/>
      <fromAddress value="tcsys@company.com" />
      <toAddress value="admin1@company.com" />
      <suppressionPeriod value="4200"/>
      <emailFormat value="html"/>
    </EmailResponder>
  </RespondersConfig>
</ApplicationConfig>
```

```

    </EmailResponder>
<EmailResponder id="EmailResponder2">
    <protocol value="smtp"/>
    <hostAddress value="svclsmtp.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" values="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>
```

Configure monitoring with the server manager administrative interface

This procedures assumes you have the server manager administrative interface running.

For information about starting the interface, see [Start the J2EE administrative interface](#).

- Under the **Administer Pool-name manager monitoring** heading, click **id=ServerManager_Monitoring_Configurations**.

This view lists the monitoring mode and all the metrics available for monitoring.

- 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.

3. In the same view, click the **Administer pool-name manager monitoring: id=EmailResponder1** value.
4. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently e-mail 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.

- **fromAddress**

Specify the address from which the notification E-mails are sent.

- **hostAddress**

Specify the server host from which the e-mail 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.

- **SuppressionPeriod**

Specify the amount of time (in seconds) to suppress e-mail notification of critical events.

For more information, see the suppression period example in [Introduction to monitoring](#).

- **toAddress**

Specify the address to which the notification E-mails are sent. You can specify multiple e-mail addresses, separated by commas.

5. Click **Apply**.
6. Click **Back to Agent View**.
7. Under the **Administer pool-name manager monitoring** heading, click **id=LoggerResponder1**.
8. (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.

For more information, see the suppression period example in [Introduction to monitoring](#).

9. Click **Apply**.
10. Click **Back to Agent View**.
11. Configure monitoring of any of the server manager metrics listed under the **Administer Pool-name manager monitoring** heading.
 - a. Click the desired metric.
For example: **id=ColdServers**.
 - b. Set the value for the **Configure_mode** attribute to one of the following:
 - **Collect**
Collect metric data and display results in the MBean view for this metric. This is the default setting.
 - **Alert**
Collect metric data, display results in the MBean view for this metric, and send e-mail notifications when critical events occur.
 - **Off**
No metric data is collected.
 - c. (Optional) To be notified when the criteria reaches the specified threshold, specify the **EmailResponder** and **LoggerResponder** values for the **Configure_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**.

Teamcenter server monitoring

Teamcenter server monitoring metrics

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive e-mail notification when specified metrics cross specified thresholds.

Metric	Description
POMRetries	Number of POM retries
Deadlocks	Number of deadlock
VeryLongRunningQueries	Number of very long running queries
DBConnectionLosses	Number of database connection losses
TimesDBOutOfSpace	Number of times the database runs out of space

Metric	Description
SqlTripCount	Number of SQL trips
DetailedSqlStats	Number of SQL statistics
SqlTotalTime	Total SQL entries
OmAllocations	Number of object manager (OM) allocations since the last call
OmCurrentAllocated	Number of total allocations for the current OM state
OsMemoryTotal	Total memory space used by the operating system
OsMemoryPeak"	Peak memory space used by the operating system
OsBsmUndoPool	Size of the BSM undo pool from the operating system
BsmUndoPoolInUse	Size of the BSM undo pool in use
PomLocks	Number of POM locks
OmModelData	Total of Object Manager (OM) model data
OmRollbackData	Total of Object Manager (OM) rollback data
Grave Events	Fatal or unexpected errors occurring in the server

Configure Teamcenter server monitoring using either:

- The **serverMonitorConfig.xml** file.
For more information about using the XML file, see [Configure monitoring with the serverMonitorConfig.xml file](#).
- The J2EE server manager administrative interface.
For more information about using the J2EE server manager administrative interface, see [Administering monitoring options for Teamcenter servers](#).

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 e-mail notifications are generated from the **TC_ROOT/pool_manager/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 monitoring with the **serverMonitorConfig.xml** file

1. Open the **TC_ROOT/pool_manager/serverMonitorConfig.xml** file.
2. 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 e-mail 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 e-mail responder. Multiple e-mail responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the e-mail protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the e-mail 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 E-mails are sent.

- **toAddress**

Specify the address to which the notification E-mails are sent.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress e-mail notification of critical events.

For more information, see the suppression period example in [*Introduction to monitoring*](#).

- **emailFormat**

Specify the format in which the e-mail 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.

For more information, see the suppression period example in [Introduction to monitoring](#).

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 server manager administrative interface) for this metric.

This is the default setting.

- **Alert**

Collect metric data, display results in the MBean view for this metric, and send e-mail notifications when critical events occur.

- **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.

Server monitoring is enabled for the metrics you configured.

Sample serverMonitorConfig.xml code

In the following example, two **EmailResponder** elements are configured. E-mail notification of deadlock critical events are sent to **admin1@company.com**, and e-mail notification of very long running queries are sent to **admin2@company.com**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Copyright 2011 Siemens Product Lifecycle Management Software Inc. All Rights Reserved.
=====
Copyright 2011.
```

```

Siemens Product Lifecycle Management Software Inc.
All Rights Reserved.
=====
Copyright 2011 Siemens Product Lifecycle Management Software Inc. All Rights Reserved.
-->
<!-- Server Health Monitoring Configuration -->
<ApplicationConfig id="TcServer" 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"/>
    <hostAddress value="svclsmtp.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="svclsmtp.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 id="Deadlocks" maxEntries="50" metricType="integer"
  mode="Alert" name="Deadlocks">
  <Threshold>
    <ThresholdValue
      description="If number of deadlocks exceeds this limit notify"
      name="NumberofDeadLocks" value="10"/>
  </Threshold>
<Responders>
  <ResponderRef id="EmailResponder1"/>
</Responders>
</Metric>
<Metric id="VeryLongRunningQueries" maxEntries="150"
  metricType="integer" mode="Alert" name="VeryLongRunningQueries">
  <ThresholdWithPeriod>
    <ThresholdValue
      description="If number of very long queries exceeds this limit notify"
      name="NoOfVeryLongRunningQueries" value="50"/>
    <ThresholdPeriod
      description="Periods for which very long queries will be monitored."
      name="TimePeriodinSec" value="600"/>
  </ThresholdWithPeriod>
<Responders>
  <ResponderRef id="EmailResponder2"/>
</Responders>
</Metric>
</MetricsConfig>
</ApplicationConfig>

```

Configure monitoring with the server manager administrative interface

This procedures assumes you have the server manager administrative interface running.

For information about starting the interface, see [Start the J2EE administrative interface](#).

- Under the **Administer Pool-name servers monitoring** heading, click **id=Server_Monitoring_Configurations**.

This view lists the monitoring mode and all the metrics available for monitoring.

- 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.

3. In the same view, click the **Administer pool-name servers monitoring:type=Configuration,id=EmailResponder1** value.

The **EmailResponder** view appears.

4. (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently e-mail 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.

- **fromAddress**

Specify the address from which the notification E-mails are sent.

- **hostAddress**

Specify the server host from which the e-mail 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.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress e-mail notification of critical events.

For more information, see the suppression period example in [Introduction to monitoring](#).

- **toAddress**

Specify the address to which the notification E-mails are sent. You can specify multiple e-mail addresses, separated by commas.

5. Click **Apply**.

6. Click **Back to Agent View**.

7. Under the **Administer pool-name servers monitoring** heading, click **id=LoggerResponder1**.

8. (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 seconds) to suppress logging of critical events to the log file.

For more information, see the suppression period example in [Introduction to monitoring](#).

9. Click **Apply**.

10. Click **Back to Agent View**.

11. Configure monitoring of any of the Teamcenter server metrics listed under the **Administer Pool-name servers monitoring** heading.

- a. Click the desired metric.

For example: **type=Configuration,id=Deadlocks**.

- b. Set the value for the **Configure_mode** attribute to one of the following:

- **Collect**

Collect metric data and display results in the MBean view for this metric. This is the default setting.

- **Alert**

Collect metric data, display results in the MBean view for this metric, and send e-mail notifications when critical events occur.

- **Off**

No metric data is collected.

- c. (Optional) To be notified when the criteria reaches the specified threshold, specify the **EmailResponder**, **LoggerResponder** values for the **Configure_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**.

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 receive 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 requested 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.
- **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.

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.

Automatic log level change

You can configure the 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.
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.

Logging level	Description
TRACE	Logs detailed information, tracing any significant step of execution. This is the most verbose logging level.

For information about working with server manager logging levels, see [Dynamically changing logging levels of business logic servers](#).

Configuring server manager logging

There are two methods available to change logging levels for the server manager.

- Use the **log4j.xml** file, stored in the **TC_ROOT/pool_manager** directory.

This method permanently changes logging levels for the server manager *after* the server manager is restarted. Changes persist until modified again in the file.

- Use the J2EE server manager administrative interface.

This method dynamically changes logging levels for the server manager until the server manager is restarted. This method is useful to test sandbox environments, as it sets logging levels temporarily.

For information about configuring logging using this interface, see [Configure server manager logging in the J2EE server manager administrative interface](#).

In the J2EE server manager administrative interface, the list of loggers is displayed under the **log4j** heading in the **Agent View**.

- log4j
 - [logger=LoggerResponder](#)
 - [logger=Process](#)
 - [logger=Task](#)
 - [logger=com.teamcenter.jeti](#)
 - [logger=com.teamcenter.jeti.JetiTreeCache](#)
 - [logger=com.teamcenter.jeti.SharedStore](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerHealthMetrics](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerInfo](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerManager](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerPoolManager](#)
 - [logger=com.teamcenter.jeti.util.AbstractMonitoring](#)
 - [logger=com.teamcenter.mld](#)
 - [logger=com.teamcenter.mld.healthmonitoring](#)
 - [logger=com.teamcenter.mld.healthmonitoring.ApplicationConfig](#)
 - [logger=com.teamcenter.mld.healthmonitoring.ConfigManager](#)
 - [logger=com.teamcenter.mld.healthmonitoring.EmailResponder](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricConfig](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricMXBeanImpl](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricManager](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricModeController](#)

Configure server manager logging in the J2EE server manager administrative interface

- In the **Agent View**, under the **log4j** heading, click the logger whose logging level you want to configure.

- o **log4j**
 - [logger=LoggerResponder](#)
 - [logger=Process](#)
 - [logger=Task](#)
 - [logger=com.teamcenter.jeti](#)
 - [logger=com.teamcenter.jeti.JetiTreeCache](#)

The logger's MBean appears.

- Within each logger MBean, change the logging level by entering any valid logging level in the **priority** box, for example, **DEBUG**.

Name	Type	Access	Value
name	java.lang.String	RO	com.teamcenter.jeti.JetiTreeCache
priority	java.lang.String	RW	DEBUG

- Click **Apply**.

The logging level for the selected logger is changed for the server manager until the server manager is restarted.

Dynamically changing logging levels of business logic servers

Use the J2EE server manager administrative interface to dynamically change logging levels for a business logic server until the user session is restarted. This method is useful to test sandbox environments, as it sets logging levels temporarily.

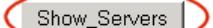
For information about configuring logging using this interface, see [Configure business logic server manager logging in the J2EE server manager administrative interface](#).

Note To make persistent changes to logging levels for all servers in the server pool, use the **logger.properties** file.

For more information, see [Configure logging with the logger.properties file](#).

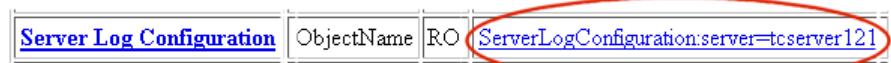
Configure business logic server manager logging in the J2EE server manager administrative interface

- In the **Agent View**, under the **Administer pool-name manager** heading, click the pool containing the server for which you want to configure logging.
- Scroll down to the **List of MBean operations** section and click **Show_Servers** to display a list of all servers in the pool.

List of MBean operations:Description of Show ServersShow_Servers

The list of servers includes each server's name, PID, status, and the user assigned to the server.

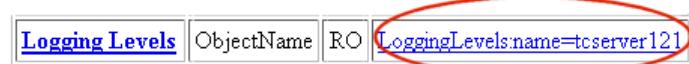
3. Click the Teamcenter server for which you want to configure logging.
4. Scroll down to the **Server Log Configuration** row and click the **ServerLogConfiguration:server=server-name@pool-name** value.



The resulting view lists the server's logging and journaling attributes and operations.

Note The first time you access this view, the selected Teamcenter server is added to the **ServerLogConfiguration** heading in the **Agent View**.

5. In this view, click **LoggingLevels:name=server-name@pool-name@machine-name**.



Note The first time you access this view, the selected Teamcenter server is added to the **LoggingLevels** heading in the **Agent View**.

6. Scroll down to the **List of MBean operations** section and click **Refresh_Loggers** to display all loggers for the Teamcenter server in the **List of MBean Attributes** table.

Description of Refresh LoggersRefresh_Loggers

7. Change the logging level of any logger.

- a. Scroll to the logger whose logging level you want to change.

- b. Type a valid logging level in the **Value** box.

Setting a logger at **DEFAULT** causes it to inherit its priority level from its parent logger.

For more information about logging levels, see [Server manager logging levels](#).

- c. Scroll to the bottom of the logging table and click **Apply**.

8. (Optional) Initialize logging for an existing logger that does not display in the logging table.
 - a. Enter a valid logger in the **Logger Name** box.
 - b. Enter a new logging level in the **Logger Value** box.
 - c. Click **Initialize_Logger**.

The new logging level is implemented for this logger. The list of all loggers for the Teamcenter server is refreshed in the **List of MBean Attributes** table.

9. (Optional) Perform any of the logging operations within the **List of MBean Operations** list to configure logging and journaling behavior.

For more information about configuring SQL logging, see [Changing SQL logging behavior](#).

To view changes to logging levels, click the Teamcenter server name under the **LoggingLevels** heading in the **Agent View**.

Changing SQL logging behavior

As the following graphic illustrates, the attributes table displays the current status of the various SQL logging settings.

Name	Type	Access	Value
SQL Logging	java.lang.Boolean	RO	false
SQL Log to the Journal file	java.lang.Boolean	RO	false
SQL Profile	java.lang.Boolean	RO	false
SQL Show Bind Variables	java.lang.Boolean	RO	false
SQL Timing	java.lang.Boolean	RO	false
TAO Log Level	java.lang.Integer	RO	0

Change the logging status using the SQL parameters. By default, the **SQL Logging** parameters display as **True**, as the following graphic illustrates.

Description of Change_SQL_Logging

Boolean Change_SQL_Logging (java.lang.Boolean) Enable	<input checked="" type="radio"/> True <input type="radio"/> False
(java.lang.Boolean) Show Bind Variables	<input checked="" type="radio"/> True <input type="radio"/> False
(java.lang.Boolean) Profile	<input checked="" type="radio"/> True <input type="radio"/> False
(java.lang.Boolean) To Journal File	<input checked="" type="radio"/> True <input type="radio"/> False
(java.lang.Boolean) Timing	<input checked="" type="radio"/> True <input type="radio"/> False

Selecting **True** or **False** for any or all of the SQL logging parameters and clicking **Change_SQL_Logging** updates the SQL logging settings on the server.

The status changes in the attributes table and the parameters are all reset to **True**.

Configuring Teamcenter server journaling

Journaling behavior determines which modules write information to the journal file as each routine is entered and exited.

1. In the **Agent View**, under the **Administer pool-name servers** heading, click the Teamcenter server for which you want to configure journaling.

This view lists the server's logging and monitoring attributes.

2. Click the **ServerLogConfiguration:server=server-name@pool-name** value.

This view lists the server's logging and journaling attributes and operations.

Note The first time you access this view, the selected Teamcenter server is added to the **ServerLogConfiguration** heading in the **Agent View**.

3. In this view, perform any of the journaling operations within the **List of MBean Operations** table to configure journaling behavior.

4. Click **ModuleJournaling:name=server-name@pool-name@machine-name**.

Note The first time you access this view, the selected Teamcenter server is added to the **ModuleJournaling** heading in the **Agent View**.

5. In this view, click **Refresh Modules** to display all journal modules for the Teamcenter server in the **List of MBean Attributes** table.

By default, journaling for each module is off. Enable journaling for any module by setting the module's value to **True**.

Alternatively, enable journaling for all modules by clicking **Activate_All_Modules**. Disable journaling for all modules by clicking **Deactivate_All_Modules**.

For subsequent changes to journaling behavior, you can click the Teamcenter server name under the **ModuleJournaling** heading in the **Agent View**.

J2EE server manager administrative interface

Using the J2EE server manager administrative interface

Note Before you can access this interface, you must complete the following tasks:

- Install the server manager.

For more information, see either the *Installation on UNIX and Linux Servers Guide* or the *Installation on Windows Servers Guide*.

- Deploy the Teamcenter Web tier application (EAR file bundling a WAR file).

For more information, see the *Web Application Deployment Guide*.

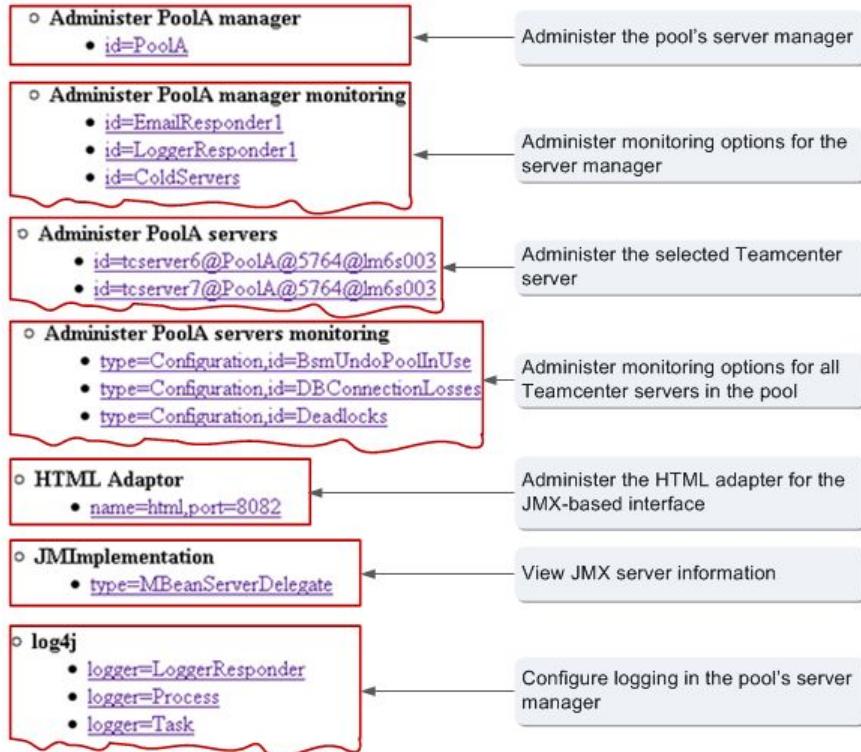
After installing and configuring the server manager, use the HTML-based server manager interface to manage the server manager tasks as shown.

Agent View

Filter by object name:

This agent is registered on the domain *DefaultDomain*.
This page contains 161 MBean(s).

List of registered MBeans by domain:



Start the J2EE administrative interface

1. Launch the J2EE server manager administrative interface from:

`http://manager-host:jmx-http-adaptor-port`

Replace *manager-host* with the machine on which the manager is running, and *jmx-http-adaptor-port* with the number of the port running a Java Management Extension (JMX) HTTP adaptor. (You define this in Teamcenter Environment Manager when you set the **JMX HTTP Adaptor Port**. By default, this value is **8082**.)

2. To log on, use the default user ID (**manager**) and password (**manager**). You can change these values using the **Change_Authentication** operation on the Pool Manager page.

The server manager displays the **Agent View** page.

Administering the pool's server manager

You can click the link below the *pool-name* **manager** MBean to display information regarding that pool. The pool page can be bookmarked for convenience.

Clicking any attribute name displays the help for that attribute. The following attribute information is available for each pool:

Global_Pool_Configuration
Host
Last Restart Warm Servers Time
Number of Assigned Servers
Number of Cold Servers
Number of Servers
Number of Warm Servers
Number of Warming Up Servers
Pool ID
Pool-Specific_Configuration
Server Pool Manager Loggers
Servers in Edit Mode
Servers in Read Mode
Servers in Stateless Mode
TreeCache_Configuration

Clicking any operation name displays the help for that operation. You can perform the following operations for any pool.

Operation	Behavior
Show_Servers	Displays a list of all servers in the pool.
Shutdown_Manager	Shuts down the server manager.
Show_Servers_Assigned_to_User	Displays a list of all servers assigned to the specified user.
Change_Global_Pool_Configuration	Changes a global pool configuration parameter dynamically. For example, use this operation to change a time out value.
Change_Pool-Specific_Configuration	Changes a pool-specific configuration parameter dynamically. For example, use this operation to change the PROCESS_TARGET .
Restart_Warm_Servers	Recycles warm servers in all server manager pools without shutting down the server manager. This is useful for updating cached values on a warm server.
	For more information, see Restarting warm servers .
Shutdown_Server	Shuts down the specified server.
Show_Pools	Displays a list of remote pools.

Operation	Behavior
Clean_Up_Pool	Cleans up data in the tree cache relating to the specified server pool.
Change.Authentication	Changes the user name and password for the server manager administration interface.
Show_TreeCache	Displays the complete contents of the tree cache.

Administering monitoring options for the server manager

All server manager monitoring options display under the **Administer server-manager-name manager monitoring** section.

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive e-mail notification when specified metrics reach specified thresholds.

- Administer PoolA manager monitoring
 - [`id=EmailResponder1`](#)
 - [`id=LoggerResponder1`](#)
 - [`id=ColdServers`](#)
 - [`id>EditModeTimeouts`](#)
 - [`id=GraveEvents`](#)
 - [`id=LoginTime`](#)
 - [`id=PoolCapacityTimeouts`](#)
 - [`id=QueryTimeouts`](#)
 - [`id=ReadModeTimeouts`](#)
 - [`id=ServerManager_Monitoring_Configurations`](#)
 - [`id=ServerManager_Monitoring_Summary`](#)
 - [`id=StatelessModeTimeouts`](#)
 - [`id=TcServerCrashes`](#)

To be notified when criteria reaches the specified threshold for any of the available metrics, set the **EmailRepsonder1** element and **LoggerRepsonder1** element at this level first, and then place the same values for the corresponding **EmailRepsonder1** and **LoggerRepsonder1** elements within the desired metrics.

For more information about the specific steps required for configuring monitoring, see [Configure monitoring with the server manager administrative interface](#).

Administering Teamcenter servers

The *pool-name servers* MBean lists all the Teamcenter servers run by the server manager. Click any link below the *pool-name servers* MBean to view server information and run server operations for that server.

Clicking any attribute name displays the help for that attribute. The following attribute information is available for each server:

- Abandoned**
- Active**
- Assigned**
- Assigned user**

Health Monitoring
Last message number
Last message time
Primary Address Host
Primary Address Port
Process ID
Ready
Server ID
Server Log Configuration
State

Clicking any operation name displays the help for that operation. You can perform the following operations for any server.

Operation name	Behavior
Show_All_Servers	Click to display a table of all servers in the pool.
Shutdown_Server	Click to shut down the server.
Show IOR	Click to display the server's IOR on a separate page.

Administering monitoring options for Teamcenter servers

All server manager monitoring options display under the **Administer pool-name servers monitoring** section.

You can configure the following metrics to provide specified levels of monitoring for specified threshold levels. Optionally, you can receive e-mail notification when specified metrics reach specified thresholds.

- Administer PoolA servers monitoring**
- [type=Configuration,id=BsmUndoPoolInUse](#)
 - [type=Configuration,id=DBConnectionLosses](#)
 - [type=Configuration,id=Deadlocks](#)
 - [type=Configuration,id=DetailedSqlStats](#)
 - [type=Configuration,id=EmailResponder1](#)
 - [type=Configuration,id=LoggerResponder1](#)
 - [type=Configuration,id=OmAllocations](#)
 - [type=Configuration,id=OmCurrentAllocated](#)
 - [type=Configuration,id=OmModelData](#)
 - [type=Configuration,id=OmRollbackData](#)
 - [type=Configuration,id=OsBsmUndoPool](#)
 - [type=Configuration,id=OsMemoryPeak](#)
 - [type=Configuration,id=OsMemoryTotal](#)
 - [type=Configuration,id=POMRetries](#)
 - [type=Configuration,id=PomLocks](#)
 - [type=Configuration,id=Server_Monitoring_Configurations](#)
 - [type=Configuration,id=SqlTotalTime](#)
 - [type=Configuration,id=SqlTripCount](#)
 - [type=Configuration,id=TimesDBOutOfSpace](#)
 - [type=Configuration,id=VeryLongRunningQueries](#)

To be notified when criteria reaches the specified threshold for any of the available metrics, set the **EmailRepsonder1** element and **LoggerResponder1** element at this

level first, and then place the same values for the corresponding **EmailRepsonder1** and **LoggerRepsonder1** elements within the desired metrics.

For more information about the specific steps required for configuring monitoring, see [Configure monitoring with the server manager administrative interface](#).

Managing the HTML adapter

Use the **HTML Adapter** MBean to manage the HTML adapter for the JMX-based server manager administrative interface.

Clicking any attribute name displays the help for that attribute. The following attribute information is available for the adapter:

Active
ActiveClientCount
AuthenticationOn
Host
LastConnectedClient
MaxActiveClientCount
Parser
Port
Protocol
ServedClientCount
State
StateString

Clicking any operation name displays the help for that operation. You can perform the following operations for any server.

Operation name	Behavior
resetParser	Removes any customization from the HtmlAdaptorServer parameter by resetting the Parser property to null.
createParser	Creates, registers and sets the Parser attribute with the HTML parser MBean.
waitForState	Specifies how long to wait for a specified state of the HtmlAdaptorServer adapter before timing out.
stop	Stops the HtmlAdaptorServer adapter.
start	Starts the HtmlAdaptorServer adapter.

Viewing JMX server details

Viewing JMX server details.

Use the **JMImplementation** MBean to view JMX server information

Clicking any attribute name displays the help for that attribute. The following attribute information is available for the adapter:

ImplementationName
ImplementationVendor
ImplementationVersion
MBeanServerId
SpecificationName
SpecificationVendor
SpecificationVersion

Configuring logging and journaling for Teamcenter servers

The **ServerLogConfiguration:server=server-name@pool-name@machine-name** MBean allows you to view and edit the configurations for system logging and journaling for the Teamcenter server session.

Clicking any attribute name displays the help for that attribute. The following attribute information is available for each server:

Checking Level
Journal Log
Journal Statistics
Journal Modules
Logging Levels
SQL Logging
SQL Log to the Journal file
SQL Profile
SQL Show Bind Variables
SQL Timing
TAO Log Level

Clicking any operation name displays the help for that operation. You can perform the following operations for any server.

Operation name	Behavior
Change_SQL_Logging	Enables SQL logging, which writes information about each SQL operation to the Teamcenter system log (syslog) file, and sets SQL logging options.
Show_Syslog	Shows the most recent portion (7 MB maximum) of the syslog file.
Show_Journal_File	Shows the most recent portion (7 MB maximum) of the file Teamcenter journal log file.
Start_Writing_Journal_Entries	Starts, or resumes, writing journal entries to the journal file.
	Journaling writes information to the journal file as each routine is entered and exited.
Stop_Writing_Journal_Entries	Stops writing journal entries to the journal file.

Operation name	Behavior
Start_Gathering_Statistics	Starts, or resumes, gathering journal statistics, adding the statistics to any previously accumulated statistics.
Stop_Gathering_Statistics	Stops gathering journal statistics, leaving previously gathered statistics in place.
Clear_Statistics	Clears accumulated journal statistics.
Dump_Statistics_To_Syslog	Writes a report of accumulated journal statistics to the syslog file.
Change_TAOLogging	Changes the TAO logging level.

For more information about the procedure, see [Configure business logic server manager logging in the J2EE server manager administrative interface](#) and [Configuring Teamcenter server journaling](#).

Configuring logging for the pool's server manager

Server manager logging behavior is controlled by a list of loggers, displayed under the **log4j** heading in the **Agent View**.

- o log4j
 - [logger=LoggerResponder](#)
 - [logger=Process](#)
 - [logger=Task](#)
 - [logger=com.teamcenter.jeti](#)
 - [logger=com.teamcenter.jeti.JetiTreeCache](#)
 - [logger=com.teamcenter.jeti.SharedStore](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerHealthMetrics](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerInfo](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerManager](#)
 - [logger=com.teamcenter.jeti.serversubpoolmanager.ServerPoolManager](#)
 - [logger=com.teamcenter.jeti.util.AbstractMonitoring](#)
 - [logger=com.teamcenter.mld](#)
 - [logger=com.teamcenter.mld.healthmonitoring](#)
 - [logger=com.teamcenter.mld.healthmonitoring.ApplicationConfig](#)
 - [logger=com.teamcenter.mld.healthmonitoring.ConfigManager](#)
 - [logger=com.teamcenter.mld.healthmonitoring.EmailResponder](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricConfig](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricMDXBeanImpl](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricManager](#)
 - [logger=com.teamcenter.mld.healthmonitoring.MetricModeController](#)

Clicking any logger in the list displays the loggers MBean.

Within each logger MBean, change the logging level by entering a valid logging level in the **priority** box and clicking **Apply**.

Managing CORBA IORs

Teamcenter uses CORBA to communicate between the Web tier and Teamcenter servers. The Web tier connects to an assigned server based on information in the CORBA interoperable object reference (IOR) produced by the Teamcenter server and sent to the server manager. Each IOR contains one or more IP addresses used to open a connection. If the primary IP address does not work, and there are no alternate addresses, the connection fails. If there are alternate addresses, they are tried one by one until a connection succeeds or all the addresses fail. Because each attempt takes time, an unusable primary IP address in the IOR (with a usable alternate address) affects four-tier performance.

Network interfaces may have different speeds. Four-tier performs better with a fast connection between the Web tier and enterprise tier machines. If the primary IP address is not the ideal address to use in connecting to the servers, modify the IP address by specifying a value for the **SERVER_HOST** parameter when configuring the server manager.

The four-tier architecture provides several diagnostics for identifying CORBA connection issues:

- The server manager logs a warning when servers that it is managing supply IORs containing multiple IP addresses.
- The Web tier confirms that the host supplied as the primary IP address in IORs for Teamcenter servers can be reached from the Web tier.
- The server manager administrative interface displays the addresses contained in the IOR for each server in the pool.

View IOR addresses

Use the server manager HTML-based administrative interface to view all addresses contained in an IOR:

1. Navigate to the server MBean page for the desired server. The primary address is shown in the attribute table.
2. Click **Show IOR** to see all addresses contained in the IOR.

.NET server manager administrative interface

Start the .NET administrative interface

1. Launch the .NET server manager administrative interface from:

`http://manager-host:port/product-name/admin`

Replace *manager-host* with the machine on which the manager is running and *port* with the number of the IIS Website port. By default, this value is **80**.

Replace *product-name* with the name of the product. By default, this value is **tc**.

2. To log on, use your operating system administrative user name and password.

Global pool configuration

Global Configuration view

Use the **Global Configuration** view to set global pool parameters, view available server manager instances, and perform global pool operations.

Note The .NET architecture provides less functionality than the J2EE architecture. To manage monitoring and logging metrics, use the J2EE architecture.

Setting global pool parameters

You can set parameters for the following global pool configuration elements:

- Soft Timeout Stateless**
- Soft Timeout Read**
- Soft Timeout Edit**
- Query Timeout**
- Hard Timeout Stateless**
- Hard Timeout Read**
- Hard Timeout Edit**
- Process Ready Timeout**

After changing a value, click **Edit** to implement the change.

Clicking any element displays help for the selected element.

Viewing server manager instances

The table following the **Server Manager Instances** heading displays all available server managers.

You can sort the table contents by clicking either the **Server Manager ID** or **State** column titles.

Server Manager ID	State ^
8085@localhost	Running
8585@localhost	Running
8080@localhost	Running

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.

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 configuration

Server manager views

Use the various server manager views to view the selected server manager's status, configure its pools, view its Teamcenter server instances, and perform server pool operations.

Note

The .NET architecture provides less functionality than the J2EE architecture. For a wider range of server manager operations, use the J2EE architecture.

Viewing server manager status

The following server manager status information is available for each server manager:

- **Server Manager ID**

Displays the ID of the server manager in *port@host*, specifying the host and port on which the server manager is running.

- **Available Servers**

Displays the number of business logic servers in **Ready** mode and available for assignment to users. If this number reaches zero, a business logic server cannot be assigned to new users on this server manager. Users already logged on receive continued access.

- **Busy Level**

Displays the percentage of business logic servers in use relative to the maximum number of servers assigned to the server manager. If there are multiple server managers, load balancing across the server managers ensures each server manager is at approximately the same level.

- **Active Status**

Displays whether the server manager is active or inactive.

You can click the **Activate** link to make the server manager inactive.

Clicking any element displays help for the selected element.

Configuring server manager pools

You can set parameters for the following global pool configuration elements:

Pool ID
Process Warm
Process Max
Process Target
Logins Per Minute

After changing a value, click **Edit** to implement the change.

Clicking any element displays help for the selected element.

Viewing Teamcenter server instances

The table following the **Business Logic Server Instances** heading displays all available Teamcenter servers.

You can sort the table contents by clicking any of the column titles.

Business Logic Server Instances							
	Server Name	State	User	Mode	Authenticated	PID	Last Message Time
<input type="checkbox"/>	tcserver38	● Ready	-	-	-	12640	5/14/2011 7:00:36 AM
<input type="checkbox"/>	tcserver39	● Ready	-	-	-	2136	5/15/2011 7:00:07 AM

Performing server manager operations

You can perform the following operations for any server:

- **Stop Server(s)**

Stops all the servers in the selected server manager view.

- **Download History Log**

Downloads the server manager history log in an Excel comma separated values (**.csv**) format.

Note

The .NET architecture provides less functionality than the J2EE architecture. For a wider range of server manager operations, use the J2EE architecture.

Using the .NET server manager administrative interface

Note

Before you can access this interface, you must complete the following tasks:

- Install the server manager.

For more information, see the *Installation on Windows Servers Guide*.

- Deploy the Teamcenter Web tier application (EAR file bundling a WAR file).

For more information, see the *Web Application Deployment Guide*.

After installing and configuring the server manager, use the .NET-based server manager interface to manage the global server pool and individual servers.

The .NET architecture provides less functionality than the J2EE architecture. To manage monitoring and logging metrics, use the J2EE architecture.

The screenshot shows two main sections of the .NET Server Manager administrative interface:

- Global Configuration:** A table with the following settings:

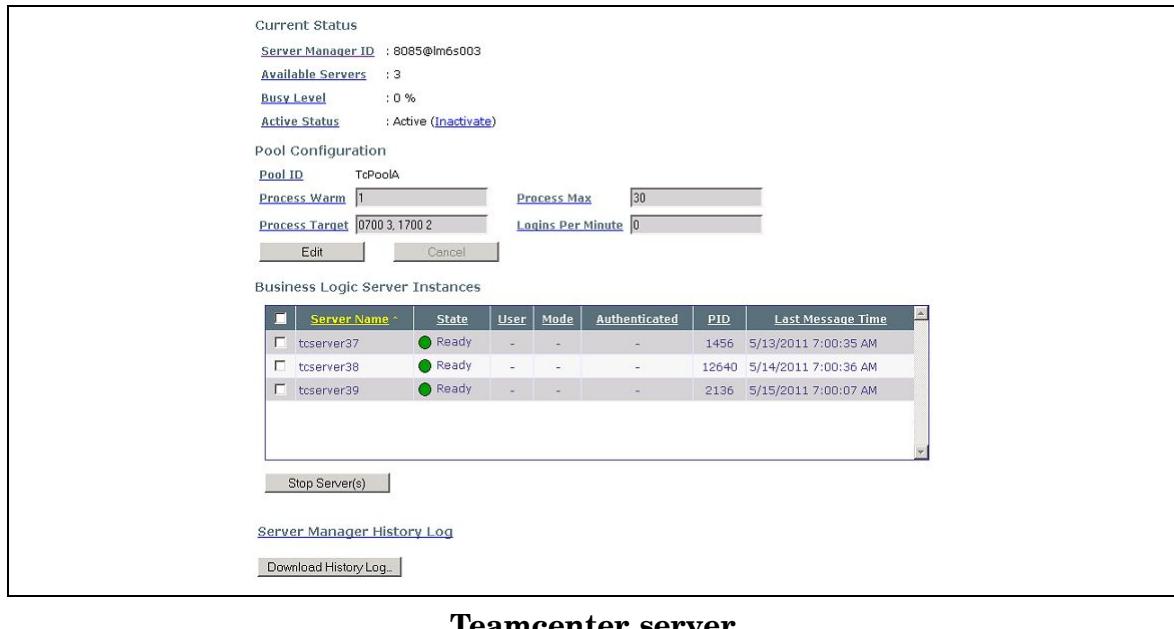
Soft Timeout Stateless (sec.)	1200	Hard Timeout Stateless (sec.)	28800
Soft Timeout Read (sec.)	28800	Hard Timeout Read (sec.)	28800
Soft Timeout Edit (sec.)	28800	Hard Timeout Edit (sec.)	28800
Query Timeout (sec.)	0	Process Ready Timeout (sec.)	300

Buttons: **Edit** and **Cancel**
- Server Manager Instances:** A table showing one instance:

Server Manager ID	State
8085@localhost	Running

Buttons: **Restart Warm Servers**

Global configuration



Teamcenter server

Using third-party applications to view server manager administration data

You can use third-party applications to view server manager administration interface data in a more comprehensive manner. For example:

- 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).

<http://download.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html>

- 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).

<http://download.oracle.com/javase/6/docs/technotes/guides/visualvm/index.html>

- 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.

http://h71000.www7.hp.com/openvms/products/openvms_ovo_agent/

Chapter

7 Updating property values in bulk

Process for updating property values in bulk	7-1
Using command line arguments to update property values in bulk	7-2
Using an XML input file to update property values in bulk	7-5
Importing updated properties back into Teamcenter	7-7
Performance statistics	7-7
Best practices and considerations	7-8

Chapter

7 *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 **attribute_export** and **txml_import** utilities in a two-step process:

1. Use the **attribute_export** 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.

For more information about performing this step, see [Using command line arguments to update property values in bulk](#) for simple updates and [Using an XML input file to update property values in bulk](#) for complex updates.

2. Run the existing **txml_import** utility with the **-bulk_load** and **-bypassSiteCheck** arguments to import the updated properties stored in the TC XML file back into the database. (If you specify any other optional arguments, they are ignored.) To use this argument, you must set a license key value for the **SITCONS_AUTH_KEY** environment variable. You must obtain the key from GTAC.

The values are updated during import.

For more information about performing this step, see [Importing updated properties back into Teamcenter](#). For more information about loading bulk data, see the [Data Exchange Guide](#).

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 **attribute_export** 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

For more information, see [Using command line arguments to update property values in bulk](#).

- 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.

For more information, see [Using an XML input file to update property values in bulk](#).

Using command line arguments to update property values in bulk

When performing a simple update (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.

- o 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.

- o 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.

- o 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.

- o 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.

- o 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.

- o 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-separate 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 **attribute_export** 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 the following command on a single line:

```
attribute_export -u=infodba -pf=d:\password.txt -g=dba  
-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:

```
attribute_export -u=infodba -p=infodba -g=dba -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 (updating many values on many properties throughout the database), use an XML input file to provide the query/update information. 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.
- Export the data to a TC XML file.

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, Tues" />
        <cond_prop attrName = "last_mod_date" cond_operator = ">" attrValue = "01-DEC-2011 00:00" />
      </where>
      <update>
        <update_prop attrName = "object_desc" attrValue = "red Desc" />
        <update_prop attrName = "int_VLA" 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)

- o 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.

- o 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.

- o 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, Tues" />
  <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

- o 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.

- o 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-separate 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
 - o Fixed length array
 - o Variable length array (VLA)

Importing updated properties back into Teamcenter

After the specified properties are queried from the database and exported along with their new values, you must import the updated properties back into Teamcenter. The values are updated in the database during import.

Run the **tcxml_import** utility with the **-bulk_load** and **-file** arguments and the **-bypassSiteCheck** switch to import the updated properties stored in the TC XML file back into the database.

The **-bypassSiteCheck** switch allows import of TC XML data into the site from where it was exported.

Performance statistics

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

Custom ADSParts	attribute_export in seconds	txml_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** preference is set to **0**.

For more information, see [Debugging using business logic server logging](#).

Best practices and considerations

Siemens PLM 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, 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 PLM Software recommends you determine the following before performing a complex bulk update of property values:

- Affected targets

Run the **attribute_export** 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 UIDs 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.

For more information about this utility, see the *Utilities Reference*.

- 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.

For more information, see *Performance statistics*.

- Run the **attribute_export** 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.

For more information about this argument, see the *Utilities Reference*.

- Run the **attribute_export** utility with the **-islandsizer** 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 island by closure rules.

The default size is 100.

For more information about this argument, see the *Utilities Reference*.

- 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"

Chapter

8 *File Management System*

Introduction to File Management System	8-1
Benefits of using FMS	8-1
FMS components	8-3
FMS server cache	8-3
FMS client cache	8-4
FMS configuration files	8-5
Introduction to FMS configuration files	8-5
FMS master configuration file	8-6
FMS server configuration file	8-7
Overview of the FMS server configuration file	8-7
General FSC configuration parameters	8-8
FSC whole file cache parameters	8-9
FSC read cache parameters	8-11
FSC write cache parameters	8-11
FSC internal cache parameters for idle file handles	8-12
FSC internal cache parameters for ticket caches	8-12
WebRAID FSC tuning parameters	8-13
FMS client configuration file	8-13
Overview of the FMS client configuration file	8-13
General FCC configuration parameters	8-14
FCC cache parameters	8-15
Sizing FMS fast cache	8-18
Overview of FMS fast cache methods	8-18
FMS fast cache fast method	8-19
Fast method for sizing the FMS fast cache	8-19
Example 1: 4 GB cache	8-20
Example 2: 40 GB cache	8-22
Refined method for sizing the FMS fast cache	8-24
FMS fast cache table method	8-26
Table method for sizing the FMS fast cache	8-26
FCC with partial mapping	8-27
FCC with no partial mapping	8-29
FSC 32-bit with partial mapping	8-31
FSC 32-bit with no partial mapping	8-33
FSC 64-bit with partial mapping	8-34
FSC 64-bit with no partial mapping	8-37
Memory considerations for FMS fast cache	8-40
Memory considerations when sizing the FMS fast cache	8-40
Example 1: 4 GB cache using fast method	8-41

Example 2: 40 GB cache using fast method	8-43
Administering FMS	8-45
Introduction to administering the FMS	8-45
Administering FSCs	8-45
Introduction to administering FSCs	8-45
Managing your FSC on Windows	8-46
Managing your FSC on UNIX	8-46
Manually configuring an FSC	8-47
Maintaining the FSC whole file cache	8-48
Copying the FSC whole file cache	8-48
Benefits of whole file cache (WFC)	8-49
Administering FCCs	8-49
Introduction to administering FCCs	8-49
Shutting down a TCCS/FCC instance	8-51
Restarting an FCC	8-53
Restart an FCC	8-53
Elements requiring a restart of an FCC	8-53
Reset a user's environment	8-54
Reconfiguring an FCC	8-55
Using the FCC assignment mode element to override default client mapping behavior	8-56
Auditing FSCs	8-57
Introduction to auditing FSCs	8-57
Enable audit logging	8-59
Audit log properties	8-59
Format specifications	8-60
Transaction identifiers (IDs)	8-64
Configuring FMS	8-65
Managing FMS host names on IBM AIX systems	8-65
Configuring FMS to run multiple versions of Teamcenter	8-67
Configuring multiuser support	8-67
Configuring FMS for HTTPS	8-68
Introduction to configuring FMS for HTTPS	8-68
Configure FMS for HTTPS	8-69
Keystores and key entries	8-70
Generate a keystore and key entry	8-71
Create a certificate signing request (CSR)	8-72
Importing certificates into the FSC keystore	8-72
Configuring native FSC client proxy in TcServer	8-73
Configuring PKI authentication	8-75
Best practices	8-75
Restricting selected fscadmin commands	8-76
Protecting the FMS encryption key	8-79
Resolving ticket expiration errors	8-82
Configuring a PAC file to run the FMS Java applet	8-82
Administering transient volumes	8-83
Introduction to administering transient volumes	8-83
Transient volume configuration components	8-83
Configuring transient volume elements in the master configuration file	8-84
Modifying the transient volume ID for the current server context	8-85
Determining the transient volume ID for a different server context	8-86
Modifying transient volume ID components	8-87

Administering volumes	8-87
Introduction to administering volumes	8-87
Default volumes	8-87
Default local volumes	8-88
Introduction to default local volumes	8-88
Enabling default local volumes	8-89
Configuring default local volumes	8-90
Moving files in batch for default local volumes	8-91
Using a default local volume with a single FSC	8-93
Using a default local volume with multiple FSCs	8-94
Using a default local volume with side caching	8-94
Best practices for configuring default local volumes	8-95
Volume failover	8-96
Configuration failover versus volume failover	8-96
Working with configuration failover	8-96
Working with volume failover	8-97
Configuring volume failover	8-99
Configuring FSC volume failover	8-99
Configuring volume failover during file import	8-100
Volume data	8-101
Volume allocation rules	8-101
Allocate volume data	8-102
DTD file of volume allocation rules	8-103
Sample volume allocation rules XML file	8-103
Moving volumes within an enterprise	8-104
Load balancing FMS data	8-105
Introduction to load balancing FMS data	8-105
Examples of load balancing FMS data	8-106
Using external hardware devices for load balancing	8-107
Introduction to using external hardware devices for load balancing	8-107
Local load balancing example	8-108
Remote load balancing example	8-110
Working with client maps	8-111
Introduction to working with client maps	8-111
Using subnet/mask attributes in a client map	8-112
Using CIDR attributes in a client map	8-112
Using domain name client maps	8-113
How the system processes client maps	8-114
Client map specificity	8-115
Accessing multiple FMS databases through a single FCC	8-115
Introduction to accessing multiple FMS databases through a single FCC	8-115
fmssmaster.xml configuration example	8-116
fcc.xml configuration example	8-117
fcc.xml personal use configuration example	8-117
Compressing FMS files	8-117
Overview of compressing files for multisite transfer	8-117
File compression example	8-118
Determining which transport method is used	8-119
Routing FSCs between sites	8-120
Accessing remote volumes using aliases (shared network)	8-122
FMS monitoring	8-124
Introduction to File Management System monitoring	8-124
Configure monitoring with the fscMonitoringConfig.xml file	8-126

Sample fscMonitorConfig.xml code	8-128
Configure monitoring with the administrative interface	8-130
Start the administrative interface	8-131
Improving cache performance	8-132
Sample FMS configurations	8-132
About the sample FMS configurations	8-132
Sample LAN configurations	8-132
Single FSC configuration	8-132
FCC direct connect configuration	8-135
FSC cached configuration	8-138
Remote user WAN configurations	8-140
FSC cached remote office configuration	8-140
Remote FSC without caching configuration	8-143
Exit FSC cache configuration	8-144
FCC LAN client failover configuration	8-147
FSC clientmap DNS suffix configuration	8-150
FCC external load balancing configuration	8-152
FSC volume load balancing of FMS data configuration	8-153
FSC volume failover configuration	8-155
FSC remote cache failover configuration	8-156
Alternate FSC remote cache failover configuration	8-159
FSC remote multiple level cache failover configuration	8-162
FSC remote multiple-level hot cache failover configuration	8-166
FSC group import multisite routing configuration	8-170
FMS shared network configuration	8-171

Chapter

8 *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 PLM Software reserves the right to change FMS behavior in the future to enhance performance or improve reliability.

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.

- Master configuration server

FMS provides the capability to administer the FMS deployment configuration with a single 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 requestor 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 requestor 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.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, analyses the configuration, computes the configuration download for each client, and provides this information as a bootstrap download when the FCC process initializes.

- Fail over (configuration dependent)

The FSC provides the capability to fail over to alternate FSCs upon failure of a specific FSC. Fail over 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 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.

- Fail over (configuration dependent)

The FCC provides the capability to fail over to alternate FSCs on failure of a specific FSC. Fail over 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:

- FMS master configuration file (**fmsmaster.xml**)
- FMS server configuration file (**fsc.xml**)
- FMS client configuration file (**fcc.xml**)

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/UNIX parameter values

Specify separate Windows and UNIX parameter values by using a vertical **or** character, for example, **\$HOME/FscCache|/tmp/FscCache**.

FMS master configuration file

The FMS master configuration file (**fmsmaster.xml**) is used to manage the FMS configuration. This file contains routing information and the FSC and FCC defaults.

A basic FMS master configuration file contains the following elements:

- **xml version**

The file begins with the standard XML statement declaring the file as an **.xml** file.

- **fmsworld**

This element is the outer containing XML element.

- **fmsenterprise**

Contains the primary configuration content including the FMS topology and the FSC and FCC parameter defaults.

- **fscdefaults**

Contains the parameter defaults for all the site FSCs, and indicates which of these parameters can be overridden by a particular FSC installation.

- **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 and thin clients.

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.xml** file.

A basic FSC configuration file contains the following elements:

- **xml version**

The file begins with the standard XML statement declaring the file as an **.xml** file.

- **fsconfig**

This element is the outer containing XML element.

- **fscdefaults** (deprecated)

Defines or overrides FSC default values from the master configuration file (**fmsmaster.xml**).

Note Placing this element in the FSC is deprecated. Siemens PLM Software recommends placing this element in the master configuration file, exposing the element to all FSCs generating configurations from the master configuration file that owns the FSC file.

- **fccdefaults** (deprecated)

Defines or overrides FCC default values from the master configuration file (**fmsmaster.xml**).

Note Placing this element in the FSC is deprecated. Siemens PLM Software recommends placing this element in the master configuration file, exposing the element to all FSCs generating configurations from the master configuration file that owns the FSC file.

- **fmsmaster**

Specifies the location from which to download FMS configuration information. The location can be either the disk location of the **fmsmaster.xml** file or the address of another FSC.

Downloaded FMS configuration information results from the merge of the **fmsmaster.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.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 fail over.

- **fsc**

Specifies the identity of the installed FSC. This identity must match an FSC defined in the master configuration file and be within an FSC group.

General FSC configuration parameters

Name	Default value	Description
FSC_LogFile	\$HOME\\${FSC_ID}.log tmp/\\$FSC_ID.log	Ignored as of Teamcenter 2007.1 changes to Log Manager functionality. See the log.properties file.
FSC_LogLevel	WARN	Defines the name of the FSC log file. The value to the left of is used for Windows hosts. The value to the right of is used for UNIX and Linux hosts.
		Defines at which log level the server runs. Valid values are FATAL , ERROR , WARN , INFO , and DEBUG .
		Warning Never run a production environment in DEBUG mode.
FSC_TraceLevel	2	Ignored. Tracing is not enabled.
FSC_EnableMonitoring	true	Not used.
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_MinimumThreads	5	Defines the minimum number of threads the server retains to service incoming requests.
FSC_MaximumThreads	255	Defines the maximum number of threads the server uses to service incoming requests.

Name	Default value	Description
FSC_MaximumThreadIdleTimeMs	10000	Defines the amount of time (in milliseconds) a thread in the service pool remains idle before it is removed and destroyed.
FSC_DelayedVolumeValidation	false	This parameter is subject to the value of the FSC_MinimumThreads parameter. 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.
FSC_SSLEnabledCiphers	None. (No restrictions are set.)	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> . Sets the specific set of SSL cipher suites required to connect to the FSC.
		Add the cipher suite names in a comma-separated list. Leave no blanks or empty spaces. For example: <code>TLS_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA</code>

FSC whole file cache parameters

Name	Default value	Description
FSC_WholeFileCacheLocation	\$HOME/FSCCache /scratch/FSCCache	Specifies the absolute path to the directory containing the FSC whole cache files. The system appends \$FSC_ID/wholefile to the specified location. Directory permissions should be limited to the authorized user ID/account.

Name	Default value	Description
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.
FSC_DiskPercentFreeGoal	30	If the number of files stored in the subdirectory exceeds the configured amount, files are automatically purged to the maximum number of files specified. Specifies the percentage of free disk space on the disk containing the whole file cache. Valid values are 0 through 100 .
FSC_MaxCacheAgeDays	3650	When free disk space falls below the specified value, files are automatically purged until the specified percentage of free disk space is reached. 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 .
FSC_WholeFilePurgePeriodMinutes	240	Any files not accessed within the specified time are purged during the next scheduled purge. 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.
FSC_WholeFilePurgeInitialWaitMinutes	0	If a background purge cycle exceeds the specified time, one more purge cycles are skipped as necessary to maintain the specified schedule. 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. For more information, see Maintaining the FSC whole file cache .

FSC read cache parameters

Name	Default value	Description
FSC_ReadCacheLocation	\$HOME\FSSCache /tmp/FSSCache	Defines the read cache location. The value is entered through Teamcenter Environment Manager during FSC installation.
FSC_MaximumReadCache FilePages	40960	Modify the value either by running Teamcenter Environment Manager in maintenance mode or by manually editing the fmsmaster.xml file.
FSC_MaximumReadCache Segments	9216	All cache directories must be on a local hard disk. Cache directories cannot be on a file share directory or a mapped drive.
FSC_ReadCacheHash BlockPages	2048	Defines maximum number of file pages in the read cache. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_MaximumReadCache ExtentFiles	3	Defines the maximum number of read cache segments. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_MaximumReadCache ExtentFileSizeMegabytes	256	Defines the maximum number of hash block pages. Each hash block contains 128 hash entries. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_ReadCache PurgePolicy	age	Defines the size (in megabytes) of read cache extent files. For more information about recommended values, see Overview of FMS fast cache methods .
		Ignored. Reserved for future use.

FSC write cache parameters

Name	Default value	Description
FSC_WriteCacheLocation	\$HOME\FSSCache /tmp/FSSCache	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.xml file.
		All cache directories must be on a local hard disk. Cache directories cannot be

Name	Default value	Description
FSC_MaximumWriteCache 40960 FilePages		on a file share directory or a mapped drive.
FSC_MaximumWriteCache 9216 Segments		Defines maximum number of file pages in the write cache. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_WriteCacheHash 2048 BlockPages		Defines the maximum number of write cache segments. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_MaximumWriteCache 3 ExtentFiles		Defines the maximum number of hash block pages. Each hash block contains 128 hash entries. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_MaximumWriteCache 256 ExtentFileSizeMegabytes		Defines the maximum number of write cache extent files. For more information about recommended values, see Overview of FMS fast cache methods .
FSC_WriteCache age PurgePolicy		Defines the size (in megabytes) of write cache extent files. For more information about recommended values, see Overview of FMS fast cache methods .
		Ignored. Reserved for future use.

FSC internal cache parameters for idle file handles

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 PLM Software recommends the value be set between 1000 to 10000 ms.

FSC internal cache parameters for ticket caches

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.

WebRAID FSC tuning parameters

Name	Default value	Description
FSC_WebRaidThreshold	32K	Defines the minimum request size for WAN acceleration. WAN acceleration is provided by WebRAID. Requests smaller than this value use traditional (unaccelerated) LAN transports.
FSC_WebRaidDownloader ConnectTimeoutMs	20000	Defines the amount of time (in milliseconds) of the initial connect timeout.
FSC_WebRaidDownloader SlotRatio	3	Tuning parameter for WebRAID internal workings. WebRAID documentation recommends this value.

WebRAID downloaders are more difficult to create and maintain than traditional LAN downloaders. You can use a cache of WebRAID downloaders to help reduce the number of instances that must be created. The following table lists the values available for WebRAID instance cache parameters.

Name	Default value	Description
FSC_MaximumIdleWebRaid Downloaders	20	Defines the maximum number of idle downloaders. An idle downloader is a downloader not currently in use that can be used on a new, incoming request.
FSC_MaximumWebRaid Downloaders	30	Defines the maximum number of WebRAID downloaders.
FSC_MaximumIdleWebRaid DownloaderAgeMs	15000	Defines the lifetime of an idle WebRAID downloader. Idle downloaders that exceed the specified timeout limit are disposed of.

FMS client configuration file

Overview of the FMS client configuration file

The FMS client configuration file (**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.

A basic FCC configuration file contains the following elements:

- **xml version**

The file begins with the standard XML statement declaring the file as an **.xml** file.

- **fccconfig**

This element is the outer containing XML element.

- **fccdefault (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

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 UNIX and Linux hosts.
FCC_LogLevel	WARNING	Defines the FCC logging level. Valid values, in order of increasing detail, are OFF , ERROR , WARNING , INFO , CONFIG , TRACE , FINE and ALL . SEVERE can be used as a synonym for ERROR .
FCC_TraceLevel	CACHE CLIENT FSC ADMIN	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 . 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.
FSC_WebRaidThreshold	32k	Defines the minimum file size or segment access required to use WAN acceleration.
FCC_MaxWANSources	8	Defines the maximum number of IP sockets used for each WAN access.
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 UNIX and Linux hosts. Note Siemens PLM Software recommends you do not set this parameter except under certain circumstances where it may be required. For more information, contact your Siemens PLM 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	250	Defines how often (in milliseconds) the FCC sends status callback messages to the client during a lengthy transport operation.

Name	Default value	Description
FCC_EnableDirectFSCRouting	true	<p>Determines whether the FCC routes data requests on volumes mounted within the local FCC group to FSCs which mount these volumes.</p> <p>If set to false, the FCC routes all requests to an assigned FSC for forwarding to the appropriate volume server.</p>
FCC_IdleTimeoutMinutes	Not applicable.	<p>This parameter is obsolete. Use the maxidletime attribute in the tccs.xml file instead; setting the idle time out for TCCS and all its contained applications, including the FCC.</p>
FCC_TransientFileFSCSource	ticketuri	<p>Determines whether to use the uniform resource identifier (URI) contained in every transient file ticket.</p> <p>Note Whether your network uses IPv6 (128-bit) or IPv4 (32-bit) addresses, use host names in URIs wherever possible so the domain name service (DNS) can determine which IP address should be used.</p> <p>If you must use IP addresses and your network uses IPv6 addresses, enclose the literal IPv6 address in square brackets, for example:</p> <p>http://[2001:db8:ffff:1:101:12ff:de13:1322]:9043/tc</p> <p>Transient file ticket URIs are defined with the Default_Transient_Server preference, which specifies a default transient file server location.</p> <p>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.</p> <p>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.</p> <p>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).</p>

FCC cache parameters

The following table lists the values available for FCC common cache parameters.

Name	Default value	Description
FCC_CacheLocation	\$HOME\FCCCACHE /tmp/FCCCACHE	<p>Defines the FCC root cache location. Each user's FCC cache is in a separate subdirectory below this location.</p> <p>The value to the left of is used for Windows hosts. The value to the right of is used for UNIX and Linux hosts.</p>

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_CachePurgeCycle	5000	Deprecated. Defined how often, in minutes, the FCC checks the size of the cache for its maximum size (high water mark) and, when reached, may purge files.
FCC_WholeFileCacheSubdirectories	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_MaximumReadCacheAge	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_MinimumReadCacheAgeMinutes	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_MaximumWriteCacheAge	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_MinimumWriteCacheAgeMinutes	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. 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.

Name	Default value	Description
FCC_MinimumBackgroundIdleTimeSeconds	5	<p>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).</p> <p>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.</p> <p>Disable this behavior by setting this parameter to 0.</p> <p>Note Setting this parameter to a very high value makes the background cache file population functionality ineffective.</p>
FCC_MaxBackgroundRetries	3	<p>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.</p> <p>Disable this behavior by setting this parameter to 0. The request is immediately discarded.</p> <p>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.</p>

The following table lists the values 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_MaxExtentFileSizeMegabytes	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. For more information about recommended values, see Overview of FMS fast cache methods .

Sizing FMS fast cache

Overview of FMS fast cache methods

The File Management System (FMS) fast cache is used by every FMS client cache (FCC) and FMS server cache (FSC). This is the only cache present in the FSC. It is the portion of the FCC where excerpted partial file data is stored.

You can use one of three sizing methods to determine your cache size requirements:

- The *fast method* provides a quick estimate of the parameters appropriate for cache sizing, with a minimum of manual calculation. Use this method for *in between* cache sizes that do not appear in the fast method sizing tables.

For more information, see [Fast method for sizing the FMS fast cache](#).

- The *refined method* provides a more complete consideration of all cache parameters. Use this method for cache sizes that approach the upper limits of the capabilities of the machine on which FMS is installed or when the assumptions documented in the *fast method* do not apply.

For more information, see [Refined method for sizing the FMS fast cache](#).

- The *table method* provides sizing parameters in table format. Determine your cache sizing requirements by comparing your environment's parameters with the parameters specified in the table method sizing tables.

For more information, see [Table method for sizing the FMS fast cache](#).

Five basic parameters are used to calculate fast cache sizing requirements. These parameters are known by different attribute names depending on the context for which the fast cache instance is being configured, but the same five parameters are represented in all cases.

Use the following parameter values to calculate your fast cache sizing requirements using any of the three sizing methods. The default values are for generic and unspecified FMS cache implementation. These settings are seldom used except in cache testing. The Teamcenter Environment Manager (TEM) installer typically calculates and sets its own defaults for FSCs and FCCs at installation, based on your input. TEM defaults are the values with which FMS initially attempts to operate. If the installer does not provide defaults, or these parameters are commented out or removed, the FSC and FCC configuration parameter default values are used.

For the default parameters, see [General FSC configuration parameters](#) and [General FCC configuration parameters](#).

Note TEM defaults override the following defaults, and the general FSC and FCC configuration parameter defaults. TEM defaults are subject to change, without notice, from version to version. TEM defaults do not use the optimized methods for calculating cache size.

Parameter	Name	Units	32-bit			64-bit			Default
			Min	Max	Default	Min	Max	Default	
Hash pages	hpages	512-byte pages	1	2097151	15	1	16777215	15	

Fast cache			32-bit			64-bit		
Parameter	Name	Units	Min	Max	Default	Min	Max	Default
File pages	fpages	512-byte pages	2	2097151	1024	2	2147482624	1024
Segments	segs	16-byte pages	1	65535	1750 UNIX	1	2147482624	1750 UNIX
					1800 Windows			1800 Windows
Extent files	files	Files	0	127000	0	0	2097151	0
Extent file size	size	Mega-bytes	16	2032	16	16	33554416	16

Apply the calculations listed in [Memory considerations when sizing the FMS fast cache](#).

FMS fast cache fast method

Fast method for sizing the FMS fast cache

The *fast method* provides a quick estimate of the parameters appropriate for cache sizing, with a minimum of manual calculation. Use this method for *in between* cache sizes that do not appear in the fast method sizing tables.

The following table lists the parameters and calculations required to calculate fast cache size using the fast method, based on a total cache size of x MB.

Parameter	Calculation
Maximum size	2032000 MB (~2 TB) in 32-bit
	33554431 MB (~32 TB) in 64-bit
Total segments	tsegs = $x * 64$
16 MB extent units	xunits = $x/16$
File pages	fpages = tsegs
	This is an absolute maximum. Do not allocate more file pages unless also adding segments or extents.
	Note Check for maximum fpages in the parameter table in Overview of FMS fast cache methods . Use the smaller of the tsegs and the maximum fpages value.
Hash pages	hpages = fpages / 12.8
	This calculation sizes the hash table with a generous hash ratio (10:1 or greater). It is unlikely that increasing the number of hash pages above this amount will increase the efficiency of the fast cache.
	Note Check for maximum hpages in the parameter table in Overview of FMS fast cache methods . Use the smaller of the calculated hpages and the upper limit.

FCC calculations

Maximum number of extent files **files** = **xunits** – 1

Parameter	Calculation
Maximum extent file size	size = 16 (units are in MB)
	Note On Windows, you can increase extent file size and reduce the number of extent files until maxfiles < 1000 (or another arbitrary directory size limit). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.
FSC calculations	
Maximum number of extent files	files = 2
Maximum extent file size	size = xunits * 8
	Round up to the next multiple of 16. Units are in MB.
	Note On 32-bit, you can add extent files and reduce the extent file size until the size of each extent file is less than the maximum (2032 MB). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.
Decrement files	
Segment calculations	
segs	segs = (xunits - (files * size / 16)) * 1024.
	Units are 16 KB segments.
	Check for maximum segs in the parameter table in <i>Overview of FMS fast cache methods</i> . You can move segments to extent files as needed to maintain segs in the desired range.
	Note Moving segments to extent files can not be performed on the HP-UX platform. HP-UX installations must work with the amount of cache that fits in the segment file.
	You can increase the number of extent files, if needed, to keep both segs and size in the desired range.

Apply the calculations listed in *Memory considerations when sizing the FMS fast cache*.

Example 1: 4 GB cache

The following example calculates fast cache size using the fast method, showing that a 4 GB cache contains 4096 MB of segment space.

Parameter	Calculation
Maximum size check	$x = 4096 < 2032000$
Total 16 MB segments	$tsegs = 4096 * 64 = 262144$
16 MB extent units	$xunits = 4096 / 16 = 256$
Maximum number of 512-byte file pages	$fpages = tsegs$
	Thus $fpages = 262144$
	This is an absolute maximum. Do not allocate more file pages unless also adding segments or extents.
	Note Check for maximum fpages in the parameter table in <i>Overview of FMS fast cache methods</i> . Use the smaller of the calculated hpages and the upper limit.

Parameter	Calculation
Metafile size check	$262144 < 2097151$
Maximum number of 512-byte hash pages	<p>hpages = fpages / 12.8</p> <p>Thus hpages = 262144 / 12.8 = 20480</p> <p>This calculation sizes the hash table with a generous hash ratio (10:1 or greater). It is unlikely that increasing the number of hash pages above this amount will increase the efficiency of the fast cache.</p>
	<p>Note Check for maximum hpages in the parameter table in Overview of FMS fast cache methods. Use the smaller of the calculated hpages and the upper limit.</p>
Hash file size check	$20480 < 2097151$
FCC calculations	
Maximum number of extent files	<p>files = xunits - 1</p> <p>Thus $256 - 1 = 255$</p>
Maximum extent file size	size = 16 (units are in MB)
Windows directory check	$255 < 1000$ (Thus, there is no directory problem in this example.)
	<p>Note On Windows, you can increase extent file size and reduce the number of extent files until maxfiles < 1000 (or another arbitrary directory size limit). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.</p>
Segments	segs = (256 - (255 * 16 / 16)) * 1024 = 1024
Segment file size check	$1024 < 65535$
Extent file size check	$16 < 2032$
FSC calculations	
Maximum number of extent files	files = 2
Maximum extent file size	<p>size = xunits * 8</p> <p>Thus size = 256 * 8 = 2048</p>
Reduce extent file size	On 32-bit, you can add extent files and reduce the extent file size until the size of each extent file is less than the maximum (2032 MB). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.
	<p>Thus files = 3</p> <p>And size = 1360 (the next multiple of 16 MB smaller than 1365 MB)</p>
Decrement files	files = 2
Segment calculations	

Parameter	Calculation
seg s (in 16 KB)	$\text{seg} = (\text{xunits} - (\text{files} * \text{size} / 16)) * 1024$ Thus $256 - (2 * 1360 / 16) * 1024 = 88064$
	Check for maximum seg s in the parameter table in Overview of FMS fast cache methods . You can move segments to extent files as needed to maintain seg s in the desired range.
	Note Moving segments to extent files can not be performed on the HP-UX platform. HP-UX installations must work with the amount of cache that fits in the segment file.
	You can increase the number of extent files, if needed, to keep both seg s and size in the desired range.
Segment file size check	$88064 > 65535$ (This must be adjusted on a 32-bit machine.)
Excess segment check	Excess is $88064 - 65535 = 22529$ 16K segments, which converts to $(22849 / 1024) = 22+$ extent units (of 16 KB each)
	Excess per extent is $22+ / 2 = 11+$ (16 MB extent units per extent file), thus you must add at least 12 16-MB units to each extent file so the segment file is no longer oversized. Add 12 16-MB units to each extent file, and subtract 24 16-MB units from the segment file.
	$24 * 1024 = 24576$ 16 KB segments
Maximum extent files size (in MB)	$\text{size} = 1360 + (12 * 16) = 1552$
Maximum number of 16 KB segments	$\text{seg} = 88064 - (24 * 1024) = 63488$
Segment file size check	$63488 < 65535$
Extent file size check	$1552 < 2032$

Apply the calculations listed in [Memory considerations when sizing the FMS fast cache](#).

Example 2: 40 GB cache

The following example calculates fast cache size using the fast method, showing that a 40 GB cache contains 40960 MB of segment space.

Parameter	Calculation
Maximum size check	$x = 40960 < 2032000$
Total 16 MB segments	$tseg = 40960 * 64 = 2621440$
16 MB extent units	$xunits = 40960 / 16 = 2560$
File pages	$fpages = tseg$
	Thus $fpages = 2621440$
	This is an absolute maximum. Do not allocate more file pages unless also adding segments or extents.
	Note Check for maximum fpages in the parameter table in Overview of FMS fast cache methods . Use the smaller of the calculated hpages and the upper limit.
Metafile size check	$2621440 > 2097151$

Parameter	Calculation
Maximum number of 512-byte file pages	2097151
Hash pages	hpages = fpages / 12.8 Thus hpages = 2097151 / 12.8 = 163839
	This calculation sizes the hash table with a generous hash ratio (10:1 or greater). It is unlikely that increasing the number of hash pages above this amount will increase the efficiency of the fast cache.
	Note Check for maximum hpages in the parameter table in <i>Overview of FMS fast cache methods</i> . Use the smaller of the calculated hpages and the upper limit.
FCC calculations	
Maximum number of extent files	files = xunits - 1 Thus, $2560 - 1 = 2559$
Maximum extent file size	size = 16 (units are in MB)
Windows directory check	$255 > 1000$ Thus no directory problem in this example for a Windows platform.)
	Note On Windows, you can increase extent file size and reduce the number of extent files until maxfiles < 1000 (or another arbitrary directory size limit). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.
File count ratio	$2559 / 1000 = 2+$ (You must round up to 3.)
Maximum extent file size (in MB)	size = 3 * 16 = 48
Maximum number of extent files	files = (xunits - 1) / 3 = 853
Decrement files	files = 852
Segments (in 16 KB)	segs = (2560 - (852 * 48 / 16)) * 1024 = 4096
Segment file size check	$4096 < 65535$
Extent file size check	$48 < 2032$
FSC calculations	
Maximum number of extent files	files = 2
Maximum extent file size	size = xunits * 8 Thus size = 2560 * 8 = 20480
Reduce extent file size	On 32-bit, you can add extent files and reduce the extent file size until the size of each extent file is less than the maximum (2032 MB). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.
	Thus files = 21
	And size = 1936 (the next multiple of 16 MB smaller than 1365 MB)
Decrement files	files = 20
Parameter	Calculation
Segment calculations	

Parameter	Calculation
seg (in 16 KB)	$\text{seg} = (\text{xunits} - (\text{files} * \text{size} / 16)) * 1024$ Thus $2560 - (20 * 1936 / 16) * 1024 = 143360$ Check for maximum segs in the parameter table in Overview of FMS fast cache methods . You can move segments to extent files as needed to maintain segs in the desired range.
	Note Moving segments to extent files can not be performed on the HP-UX platform. HP-UX installations must work with the amount of cache that fits in the segment file.
Segment file size check	$143360 > 65535$ (This must be adjusted on a 32-bit machine.)
Excess segment check	$143360 - 65535 = 77825$ 16K segments, which converts to $(59905 / 1024) = 76+$ extent units of 16 KB each.
	Excess per extent is $76+ / 20 = 3+$ (16 MB extent units per extent file), thus you must add at least 4 16 MB units to each of the 20 extent file so the segment file is no longer oversized. Then subtract 80 16 MB units from the segment file.
	$80 * 1024 = 81920$ 16 KB segments
Maximum extent files size (in MB)	$\text{size} = 1936 + (4 * 16) = 2000$
Maximum number of 16 KB segments	$\text{seg} = 143360 - (80 * 1024) = 61440$
Segment file size check	$61440 < 65535$
Extent file size check	$2000 < 2032$

Apply the calculations listed in [Memory considerations when sizing the FMS fast cache](#).

Refined method for sizing the FMS fast cache

The *refined method* provides a more complete consideration of all cache parameters. Use this method for cache sizes that approach the upper limits of the capabilities of the machine on which FMS is installed or when the assumptions documented in the *fast method* do not apply.

The following table lists the parameters and calculations required to calculate fast cache size using the refined method, based on a total cache size of x MB.

Parameter	Calculation
Maximum size	2032000 MB (~2 TB) in 32-bit 33554431 MB (~32 TB) in 64-bit
Total segments	$\text{tseg} = x * 64$
16 MB extent units	$\text{xunits} = x/16$

Parameter	Calculation
File pages	<p>File pages are relatively inexpensive, and it is important that they are always available. A single 512-byte file page can hold references for up to 35 segments (UNIX) or 36 segments (Windows). If the file page is full, this represents less than a tenth of a percent of the cache size.</p> <p>The fast method uses the calculation of fpages = tsegs. This calculation provides the maximum number of file pages that may be needed for the number of segments calculated. This is sufficient and affordable for small and moderate-sized caches. For larger caches, use the maximum.</p> <p>This is an absolute maximum. Do not allocate more file pages unless also adding segments or extents.</p> <p>Note Check for maximum fpages in the parameter table in <i>Overview of FMS fast cache methods</i>. Use the smaller of the calculated hpages and the upper limit.</p>
Hash pages	<p>Use hash pages to look up file pages corresponding to a file GUID. Each hash page holds 128 hash bins. Siemens PLM Software recommends industry standard hash ratios (hratio) of 4:1 to 10:1 for standard hashing algorithms.</p> <p>Thus, hpages = fpages * hratio / 128</p> <p>Note Check for maximum hpages in the parameter table in <i>Overview of FMS fast cache methods</i>. Use the smaller of the calculated hpages and the upper limit. If calculated correctly, it is unlikely you will exceed the maximum hash file size unless you assume a particularly excessive hash ratio (greater than 128:1).</p>
FCC calculations	
Maximum number of extent files	<p>The FCC creates complete extent files as they are needed. If the extent files are large, creating them can take a significant amount of time, causing the FCC client to appear to stall while the extent file is created. Siemens PLM Software recommends you configure the FCC with relatively small extent files.</p> <p>The use of small extent files results in a large number of extent files created in the cache directory. Directory lookup performance degrades when many files with similar names are in the same directory, particularly on Windows. To prevent poor performance, limit the number of extent files to some arbitrary limit. Siemens PLM Software recommends no more than 1000 extent files per cache instance, whenever this is possible.</p> <p>files = xunits - 1</p> <p>size = 16 (Units are in MB.)</p> <p>Note On Windows, you can increase extent file size and reduce the number of extent files until maxfiles < 1000 (or another arbitrary directory size limit). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.</p>
segs (in 16 KB segments)	<p>The refined method calculations size the segment file as if it were another extent file. However, unlike extent files, the segment file is completely and continuously memory-mapped. Therefore, data stored in the segment file is accessed significantly faster than extent data. Siemens PLM Software recommends putting as much of the segment data in the segment file as memory considerations allow.</p> <p>segs = size * 64</p>
FSC calculations	

Parameter	Calculation
seg s (in 16 KB segments)	Unlike the FCC, the FSC fast cache appends to the end of existing extent files in 16-MB sections. This limits the maximum delay during any single transaction. As the cache grows it causes more, but smaller, delays. A single extent file is ideal, except you must have some data in the segment file as well. Thus, start with two, and decrement the count when you move some of that data into the segment file.
Maximum number of extent files	files = 2
Maximum extent file size	size = xunits * 8 Round up to the next multiple of 16. Units are in MB.
	<p>Note On 32-bit, you can add extent files and reduce the extent file size until the size of each extent file is less than the maximum (2032 MB). The size parameter must remain a multiple of 16. Round down to achieve the desired multiple of 16.</p>
Decrement files	
seg s (in 16 KB segments)	seg s = (xunits - (files * size * 16)) * 1024
Maximum segment check	Check for maximum seg s in the parameter table in Overview of FMS fast cache methods . You can move segments to extent files as needed to maintain seg s in the desired range. You can increase the number of extent files, if needed, to keep both seg s and size in the desired range.

Apply the calculations listed in [Memory considerations](#).

FMS fast cache table method

Table method for sizing the FMS fast cache

The *table method* provides sizing parameters in table format. Determine your cache sizing requirements by comparing your environment with the parameters specified in the example tables.

The following operating system table lists the supported operating systems, indicates whether that system maps only portions of a file when requested (partial mapping), and indicates whether the system allows extent files. The ability to map only portions of a file upon request improves system resource considerably. For example, the fast cache memory maps 16 KB portions of each extent file. Some systems memory map the entire file, returning a pointer to the requested mapped region. This method uses more resources, directly affecting your cache sizing requirements.

Supported operating system	Partial mapping	Allows extent files
Windows	Yes	Yes
AIX		Yes
Sun		Yes
Linux	Yes	Yes
HP-RISC	No	No
HP-Itanium	No	No
Macintosh		

Note

This table method does not support the HP-UX platform. This platform does not support partial mapping, and it fails when the same file (or section of file) is memory-mapped a second time by the same process. These limitations effectively prevent the fast cache from mapping extent files. Thus, on this platform, the number of extent files must be set to **0** to prevent failures and the total amount of fast cache on the machine (including hash files, metafile, and segment space) is limited to the amount of **shmem** available in the operating system kernel.

Refer to the HP-UX example tables for information on sizing the fast cache when running on a HP-UX platform.

FCC with partial mapping

The following table lists sizing requirements for an FCC, 128 MB shareable memory per user, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	12785	16384	4096	12	16	127
2048 (2 GB)	9496	121575	1024	127	16	128
16384 (16 GB)	9496	121575	1024	512	32	128
65536 (64 GB)	9496	121575	1024	819	80	128
66496 (64.9 GB) max	9496	121575	1024	831	80	128

The following table lists sizing requirements for an FCC, 256 MB shareable memory per user, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	16384	16384	11264	5	16	241
2048 (2 GB)	35797	131072	7168	121	16	242
16384 (16 GB)	28492	364722	1024	512	32	256
131072 (128 GB)	26118	334329	2048	910	144	256
199488 (194.8 GB) max	28492	364722	1024	959	208	256

The following table lists sizing requirements for an FCC, 512 MB shareable memory per user, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	88989	131072	22528	106	16	508
16384 (16 GB)	66484	851018	1024	512	32	512
131072 (128 GB)	64110	820624	2048	910	144	512
262144 (256 GB)	64110	820624	2048	964	272	512
199488 (194.8 GB) max	66484	851018	1024	970	480	512

The following table lists sizing requirements for an FCC, 1 GB shareable memory per user, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	131072	131072	53248	76	16	1009
16384 (16 GB)	166700	1048576	23552	501	32	1010
131072 (128 GB)	140094	1793216	2048	910	144	1024
524288 (512 GB)	140094	1793216	2048	993	528	1024
981120 (958.1 GB) max	140094	1793216	2048	989	992	1024

The following table lists sizing requirements for an FCC, 1.5 GB shareable memory per user, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
2048 (2 GB)	131072	131072	65535	65	16	1200
16384 (16 GB)	273083	1048576	53248	973	16	1526
131072 (128 GB)	236991	2097151	21504	908	144	1524
1048576 (1 TB)	236991	2097151	21504	993	1056	1524
1147728 (1.1 TB) max	236991	2097151	21504	996	1152	1524

FCC with no partial mapping

The following table lists sizing requirements for an FCC, 128 MB shareable memory per user, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	4823	16384	4096	12	16	123
2048 (2 GB)	9496	121575	1024	127	16	128
16384 (16 GB)p	2374	30393	1024	512	32	128

The following table lists sizing requirements for an FCC, 256 MB shareable memory per user, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	13015	16384	12288	4	16	255
2048 (2 GB)	18111	131072	8192	120	16	249
16384 (16 GB)	21369	273542	1024	512	32	256
32768 (32 GB)	11871	151967	2048	692	48	256
47984 (46.9 GB) max	11871	151967	2048	999	48	256

The following table lists sizing requirements for an FCC, 512 MB shareable memory per user, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	34495	131072	23552	105	16	497
16384 (16 GB)	59361	759837	1024	512	32	512
65536 (64 GB)	35616	455903	2048	819	80	512
111920 (109.3 GB) max	21369	273542	2048	999	112	512

The following table lists sizing requirements for an FCC, 1 GB shareable memory per user, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	67263	131072	55296	74	16	1009
16384 (16 GB)	105005	1049600	22528	502	32	1012
131072 (128 GB)	93106	1063773	2048	910	144	1024
239792 (234.2 GB) max	40365	516690	2048	999	240	1024

The following table lists sizing requirements for an FCC, 1.5 GB shareable memory per user, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	100031	131072	65535	65	16	1185
16384 (16 GB)	137773	1049600	54272	972	316	1476
131072 (128 GB)	159090	2036364	2048	910	144	1536

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per user)
262144 (256 GB)	102102	1306920	2048	964	272	1536
367664 (359 GB) max	59361	759837	2048	999	368	1536

FSC 32-bit with partial mapping

The following table lists sizing requirements for a 32-bit FSC, 128 MB shareable memory per cache, partial mapping supported.

Note There are two fast caches for each FSC: read cache and write cache. You can surpass the read cache's memory consumption recommendations in these tables if you reduce the write cache by an equivalent amount.

For example, a read cache with 768 MB memory with a write cache of 256 MB memory consumes a total of approximately 1 GB memory. The total memory consumption should allow space for the FSC process code, JRE, and cache memory usage that does not exceed the process or machine limits. This is 2 GB for 32-bit processes.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	16384	16384	6144	1	160	113
2048 (2 GB)	15850	131072	2048	1	2016	120
16384 (16 GB)	9496	121575	1024	9	1824	128
65536 (64 GB)	9496	121575	1024	33	2000	128
66544 (65 GB) max	9496	121575	1024	33	2016	128

The following table lists sizing requirements for a 32-bit FSC, 256 MB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	16384	16384	13312	1	48	241
2048 (2 GB)	42460	131072	9216	1	1904	245
16384 (16 GB)	28492	364722	1024	9	1824	256

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
131056 (128 GB)	28492	364722	1024	65	2016	256
199600 (194.9 GB) max	28492	364722	1024	99	2016	256

The following table lists sizing requirements for a 32-bit FSC, 512 MB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	95637	131072	24576	1	1664	511
16384 (16 GB)	66484	851018	1024	9	1824	512
131056 (128 GB)	66484	851018	1024	65	2016	512
262144 (256 GB)	66484	851018	1024	129	2032	512
467376 (456.4 GB) max	66484	851018	1024	230	2032	512

The following table lists sizing requirements for a 32-bit FSC, 768 MB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	131072	131072	38912	1	1440	753
16384 (16 GB)	113508	1048576	9216	9	1824	760
131056 (128 GB)	104476	1337315	1024	65	2016	768
524288 (512 GB)	104476	1337315	1024	259	2032	768
731536 (714.4 GB)	104476	1337315	1024	360	2032	768

FSC 32-bit with no partial mapping

The following table lists sizing requirements for a 32-bit FSC, 128 MB shareable memory per cache, no partial mapping supported.

Note There are two fast caches for each FSC: read cache and write cache. You can surpass the read cache's memory consumption recommendations in these tables if you reduce the write cache by an equivalent amount.

For example, a read cache with 768 MB memory with a write cache of 256 MB memory consumes a total of approximately 1 GB memory. The total memory consumption should allow space for the FSC process code, JRE, and cache memory usage that does not exceed the process or machine limits. This is 2 GB for 32-bit processes.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1797	17408	1024	8	32	122
2048 (2 GB)	2374	30393	1024	64	32	128
16384 (16 GB) max	2374	30393	1024	512	32	128

The following table lists sizing requirements for a 32-bit FSC, 256 MB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	3845	17408	3072	4	64	251
2048 (2 GB)	7123	91180	1024	32	64	256
16384 (16 GB)	7123	91180	1024	256	64	256
32768 (32 GB)	7123	91180	1024	512	64	256
47968 (46.9 GB) max	14245	182362	1024	999	48	256

The following table lists sizing requirements for a 32-bit FSC, 512 MB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	2374	30393	1024	13	160	512
16384 (16 GB)	2374	30393	1024	103	160	512
65536 (64 GB)	9496	121575	1024	455	144	512
111904 (109.3 GB) max	23743	303935	1024	999	112	512

The following table lists sizing requirements for a 32-bit FSC, 768 MB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	4747	60788	1024	9	240	768
16384 (16 GB)	4747	60788	1024	69	240	768
131072 (128 GB)	18994	243148	1024	631	208	768
175840 (171.7 GB) max	33241	425509	1024	999	176	768

FSC 64-bit with partial mapping

The following table lists sizing requirements for a 64-bit FSC, 1 GB shareable memory per cache, partial mapping supported.

Note

There are two fast caches for each FSC: read cache and write cache. You can surpass the read cache's memory consumption recommendations in these tables if you reduce the write cache by an equivalent amount.

For example, a read cache with 768 MB memory with a write cache of 256 MB memory consumes a total of approximately 1 GB memory. The total memory consumption should allow space for the FSC process code, JRE, and cache memory usage that does not exceed the process or machine limits. This is 2 GB for 32-bit processes.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	131072	131072	55296	1	1184	1009
16384 (16 GB)	173349	1048576	25600	1	15984	1013
131072 (128 GB)	147217	1884398	1024	1	131056	1024
524288 (512 GB)	147217	1884398	1024	1	524272	1024
1030528 (~1 TB) max	147217	1884398	1024	1	1030512	1024

The following table lists sizing requirements for a 64-bit FSC, 2 GB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	131072	131072	120832	1	160	2033
16384 (16 GB)	386116	1048576	84992	1	15056	2045
131072 (128 GB)	299185	3829582	1024	1	131056	2048
1048576 (1 TB)	299185	3829582	1024	1	1048560	2048
2094288 (~2 TB) max	299185	3829582	1024	1	2094272	2048

The following table lists sizing requirements for a 64-bit FSC, 4 GB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	811650	1048576	202752	1	13216	4093
131072 (128 GB)	603120	7719950	1024	1	131056	4096

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1048576 (1 TB)	603120	7719950	1024	1	1048560	4096
2097152 (~2 TB)	603120	7719950	1024	1	2097136	4096
4221840 (~4 TB) max	603120	7719950	1024	1	4221824	4096

The following table lists sizing requirements for a 64-bit FSC, 8 GB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	1048576	1048576	456704	1	9248	8177
131072 (128 GB)	1433343	8388608	216064	1	127696	8188
1048576 (1 TB)	1210990	15500688	1024	1	1048560	8192
8388608 (8 TB)	1210990	15500688	1024	1	8388592	8192
8476928 (8.1 TB) max	1210990	15500688	1024	1	8476912	8192

The following table lists sizing requirements for a 64-bit FSC, 16 GB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	1048576	1048576	980992	1	1056	16369
131072 (128 GB)	3135479	8388608	687104	1	120336	16379
1048576 (1 TB)	2426730	31062164	1024	1	1048560	16384
8388608 (8 TB)	2426730	31062164	1024	1	8388592	16384

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
16777216 (16 TB)	2426730	31062164	1024	1	16777200	16384
16987120 (16.2 TB) max	2426730	31062164	1024	1	16987104	16384

The following table lists sizing requirements for a 64-bit FSC, 32 GB shareable memory per cache, partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	81920	1048576	1048576	0	16	16937
131072 (128 GB)	6539751	8388608	1629184	1	1105616	32762
1048576 (1 TB)	4858211	62185115	1024	1	1048560	32768
8388608 (8 TB)	4858211	62185115	1024	1	8388592	32768
33554416 (32 TB) max	4858211	62185115	1024	1	33554400	32768

FSC 64-bit with no partial mapping

The following table lists sizing requirements for a 64-bit FSC, 1 GB shareable memory per cache, no partial mapping supported.

Note There are two fast caches for each FSC: read cache and write cache. You can surpass the read cache's memory consumption recommendations in these tables if you reduce the write cache by an equivalent amount.

For example, a read cache with 768 MB memory with a write cache of 256 MB memory consumes a total of approximately 1 GB memory. The total memory consumption should allow space for the FSC process code, JRE, and cache memory usage that does not exceed the process or machine limits. This is 2 GB for 32-bit processes.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	7123	91180	1024	7	320	1024
16384 (16 GB)	7123	91180	1024	52	320	1024
131072 (128 GB)	21369	273542	1024	456	288	1024
524288 (512 GB)	78357	1002986	1024	3277	160	1024
997136 (973.8 GB) max	142468	1823611	1024	62320	16	1024

The following table lists sizing requirements for a 64-bit FSC, 2 GB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	2374	30393	1024	4	672	2048
16384 (16 GB)	9496	121575	1024	25	656	2048
131072 (128 GB)	23743	303935	1024	211	624	2048
1048576 (1 TB)	151966	1945185	1024	3121	336	2048
2060816 (~2 TB) max	294436	3768795	1024	128800	16	2048

The following table lists sizing requirements for a 64-bit FSC, 4 GB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	7123	91180	1024	13	1344	4096

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
131072 (128 GB)	21369	273542	1024	100	1312	4096
1048576 (1 TB)	156715	2005972	1024	1041	1008	4096
4188432 (~4 TB) max	598371	7659163	1024	261776	16	4096

The following table lists sizing requirements for a 64-bit FSC, 8 GB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	9496	121575	1024	7	2704	8192
131072 (128 GB)	23743	303935	1024	50	2672	8192
1048576 (1 TB)	151966	1945185	1024	440	2384	8192
8388608 (8 TB)	1199117	15348722	1024	262144	32	8192
8443664 (~8.1 TB) max	1206241	15439901	1024	527728	16	8192

The following table lists sizing requirements for a 64-bit FSC, 16 GB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	7123	91180	1024	4	5440	16384
131072 (128 GB)	21369	273542	1024	25	5408	16384
1048576 (1 TB)	156715	2005972	1024	206	5104	16384
8388608 (8 TB)	1203866	15409509	1024	3049	2752	16384

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
16777216 (16 TB)	2400611	30727835	1024	262144	64	16384
16953872 (16.2 TB) max	2421981	31001377	1024	1059616	16	16384

The following table lists sizing requirements for a 64-bit FSC, 32 GB shareable memory per cache, no partial mapping supported.

Cache size MB	Hash pages	File pages	Segments	Extent files	Extent file size MB	Memory used (MB per cache)
1	5	64	64	0	16	2
32	160	2048	2048	0	16	34
256	1280	16384	16384	0	16	265
2048 (2 GB)	10240	131072	131072	0	16	2118
16384 (16 GB)	81920	1048576	1048576	0	16	16937
131072 (128 GB)	23743	303935	1024	13	10864	32768
1048576 (1 TB)	151966	1945185	1024	100	10576	32768
8388608 (8 TB)	1206241	15439901	1024	1023	8208	32768
33554416 (32 TB) max	4796474	61394884	1024	233017	144	32768

Memory considerations for FMS fast cache

Memory considerations when sizing the FMS fast cache

Every client or server on a system receives maximum memory benefit while the amount of memory consumed by the cache does not exceed the amount of memory available on the system for file memory mapping.

On a single-user Windows workstation, the amount of memory available for file memory mapping is easy to calculate. On multiuser workstations, or in environments with multiple caches, the system may virtually map the memory mapping files, giving each process more apparent memory than it actually has.

It is important to ensure your system has sufficient *swap space* (or temporary storage) to virtualize efficiently. If your system does not virtualize the mappable memory, the mappable memory may need to be proportioned (shared) among the maximum number of consumers. For example, on a 64-bit machine with 10 GB of RAM and 15 users, each user gets about 660 MB of mappable memory.

The following table lists the amount of shared memory required per FMS fast cache parameter.

File	Size parameter	Shared memory used
Hash file	hpages	$(\text{hpages} + 1) * 512$
Metafile	fpages	$(\text{fpages} + 1) * 512$
Segment file	segs	segs * 16384
Extent files ¹	size, files	Smallest of (files and 3) * 16384
Extent files ²	size, files	Smallest of (files and 3) * size * 1048576
Extent files ³	HP-UX	Extent files not supported.

¹ On systems that allow partial mapping, up to three 16384-byte windows are mapped to extent files.

² On some systems, the entire extent file is memory mapped, regardless of the partial mapping requested by FMS.

³ On HP-UX (including 64-bit Itanium systems), extent files are not supported. HP-UX does not allow an application to unmap and remap the same extent file more than once in the same process. The system may display a **FMSC_ERROR_UTIL_BAD_VM_POINTER** message due to a bad virtual memory pointer returned by the system **mmap()** library call. Given this limitation, the only functionality the fast cache offers for this platform is an exclusive segment file operation, for a maximum of 1 GB of fast cache.

Example 1: 4 GB cache using fast method

The following example lists memory considerations when calculating fast cache size using the fast method for a 4 GB cache with 1 GB of shareable memory.

File	Shared memory used FSC			Shared memory used FCC		
	Size		Size		Size	
Hash file	$(20480 + 1) * 512$	10486272	10+ MB	$(20480 + 1) * 512$	10486272	10+ MB
Metafile	$(262144 + 1) * 512$	134218240	128+ MB	$(262144 + 1) * 512$	134218240	128+ MB
Segment file	63488 * 16384	1040187392	992 MB	1024 * 16384	16777216	16 MB
Extent files ¹	2 * 16384	32768	32 KB	3 * 16384	49152	48 KB
Extent files ²	2 * 1552 * 1048576	3254779904	3104 MB	3 * 16 * 1048576	50331648	48 MB
Total¹		1184924672	1130+ MB		161530880	154+ MB
Total²		4439671808	4234+ MB		211813376	202+ MB

¹ On systems that allow partial mapping, up to three 16384-byte windows are mapped to extent files.

² On some systems, the entire extent file is memory mapped, regardless of the partial mapping requested by FMS.

In this example, the memory in both the FSC cases exceed the memory available on the machine (1 GB = 1024 MB). Additionally, each FSC has two fast caches, the read cache and the write cache. To correct for this, you must split each of the caches into more extent files.

Two factors take the cache memory consumption over the limit: the segment file and the extent files. Compute how much memory you have to spare, then split that

amount evenly among these two factors. For example, if the FSC process consumes 128 MB of memory just for the JRE and the FSC code and data areas, subtract this amount, and the two hash files, and the two metafiles from the 1 GB of available memory:

$$1024 \text{ MB} - 128 \text{ MB} (\text{JRE/FSC}) - 20 \text{ MB} (\text{hash files}) - 256 \text{ MB} (\text{metafiles}) = \text{less than } 620 \text{ MB}$$

Allocate half of the available memory (approximately 300 MB) for each cache. To allow for situations in which the entire extent file is memory mapped, allocate half of each cache's ration (150 MB) to the three extent files that may be memory-mapped at any given time. This provides for three 50 MB extent files. Each extent file needs to be a multiple of 16 MB, so the files can be either 48 MB or 64 MB. Either size has only a slight affect on the size of the segment file, but 48 MB noticeably reduces the number of extent files. It is better to choose the larger size, 64 MB. This results in $(300 - (3 * 64)) = 108$ MB of memory into which the segment file can be mapped. Therefore, the maximum extent file size is 64 MB, and the Maximum number of 16 KB segments is $108 * 64 = 6912$.

Use these figures to calculate the maximum number of extent files:

$$\text{files} = ((4096 \text{ MB} - 108 \text{ MB}) / 64 \text{ M}) = 62$$

Taken into account all of these memory considerations revises the above fast cache calculations as follows.

File	Size	Shared memory used FSC
Hash file	$(20480 + 1) * 512$	10486272
Metafile	$(262144 + 1) * 512$	134218240
Segment file	$63488 * 16384$	113246208
Extent files ¹	$3 * 16384$	49152
Extent files ²	$3 * 64 * 1048576$	201326592
Total¹	257999872	246+ MB
Total²	459277312	438+ MB

¹On systems that allow partial mapping:

JRE/FSC	128 MB
Read cache	246+ MB
Write cache	246+ MB
Total	620 MB

²On systems in which the entire extent file is mapped, regardless of the partial mapping requested by FMS:

JRE/FSC	128 MB
Read cache	438+ MB
Write cache	438+ MB
Total	1004 MB

On HP-UX (including 64-bit Itanium) systems, in which extent files are not supported, the cache must fit in the available memory, thus, **available** = $1024 \text{ MB} - 128 \text{ MB} (\text{JRE/FSC}) = 896 \text{ MB}$. In this situation, memory consummation must be less

than 448 MB per cache. Because there are no extent files, this is accomplished by reducing the number of segments, which effectively reduces the size of the cache. Resize the segment file to 433 MB. (433 MB is approximately 97 percent of 448 MB, rounded down. The other 3 percent account for the memory consumed by the resized metafile and the hash file).

The calculations for an HP-UX system are listed in the following table:

File	Size	Shared memory used FSC
Hash file	(2165 + 1) * 512	1108992
Metafile	(27712 + 1) * 512	14189056
Segment file	27712* 16384	454033408
Extent files	0	0 KB
Total	469331456	447+ MB
JRE/FSC	128 MB	
Read cache	447+ MB	
Write cache	447+ MB	
Total	1023 MB	

Note

On HP-UX systems, memory restrictions can only be addressed by reducing the size of the fast cache. In this example, the fast cache must be reduced significantly, by approximately 89%, from 4 GB to 433 MB).

Example 2: 40 GB cache using fast method

The following example lists memory considerations when calculating fast cache size using the fast method for a 40 GB cache with 1 GB of shareable memory.

File	Size	Shared memory used FSC	Size	Shared memory used FCC
Hash file	(163839 + 1) * 512	83886080	80 MB	(163839 + 1) * 512
Metafile	(2097151 + 1) * 512	1073741824	1024 MB	(2097151 + 1) * 512
Segment file	61440 * 16384	1006632960	960 MB	4096 * 16384
Extent files ¹	3* 16384	49152	48 KB	3 * 16384
Extent files ²	3* 2000 * 1048576	6291456000	6000 MB	3 * 48 * 1048576
Total¹		2164310016	2064+ MB	1224785920
Total²		8455716864	8064+ MB	1375731712

¹ On systems which allow partial mapping, up to three 16384-byte windows are mapped to extent files.

² On some systems, the entire extent file is memory mapped, regardless of the partial mapping requested by FMS.

In this example, the memory in both FSC cases exceed the memory available on the machine (1 GB = 1024 MB). Additionally, each FSC has two fast caches, the read cache and the write cache. To correct for this, you must split each of the caches into more extent files.

In this example, the metafile is considered. You can reduce metafile size by reducing the number of file pages. With caches this large, you can assume an average file size,

and use that assumption to compute an average number of file headers per file. For example, let **average** represent the average file size, in bytes, understanding that some files are actually larger, and some smaller:

Calculation	Definition
filesegs = average / 16384	The average file consumes some number of segments.
	Round up.
filepages = filesegs / 35 (UNIX)	The average file consumes some number of file pages.
filepages = filesegs / 36 (Windows)	Round up.
averagerefs = filesegs / filepages	The average file page is typically only partially full.
	Round down.
fpages = segs / averagerefs	Fewer file pages are required to serve the larger files.
	Round up.

Note

If you expect to fill the 40 GB cache with very small files (smaller than 16 KB) you need all of this metafile space to track all the individual files. Therefore, you need more memory on your machine, or accept working with a smaller cache.

Assuming a file size of 1.4 MB results in the following calculations:

Calculation	Definition
average = 1468006	Average bytes per file.
filesegs = 1468006 / 16384 = 89.6	Average file segments per file.
	Round up to 90.
filepages = 90 / 35 (UNIX) = ~2.6	Average file pages per file.
filepages = 90 / 36 (Windows) = ~2.6	Round up to 3.
averagerefs = 90 / 3 = 30	Average segment references per file page.
	Round down.
fpages = 2621440 / 30 = 87382	Required number of file pages.

These computations recommend a 97 percent reduction in file pages. Siemens PLM Software recommends a 90 percent reduction, which provides the additional benefit of handling unanticipated shifts in the average file size as your environment ages. File pages are relatively inexpensive, and you never want to run out. As long as 15–20 segment references are used per file page, on average, full segment space capacity is supported in the cache. Therefore, file pages required is $2621440 * 10\% = 262144$, and Hash pages required is $262144 / 12.8 = 20480$.

The calculations for a 40 GB fast cache, with 1 GB of shareable memory and an average file size of 1.4 MB are:

File	Size	Shared memory used FSC	Size	Shared memory used FCC
Hash file	$(20480 + 1) * 512$	10486272	10+ MB	$(20480 + 1) * 512$
Metafile	$(262144 + 1) * 512$	134218240	128+ MB	$(262144 + 1) * 512$

Next, recalculate the available memory, decrease the number of segments, and increase the number of extent files to bring the segment and extent files into alignment.

For more information on these steps, see [Example 1: 4 GB cache using fast method](#).

Administering FMS

Introduction to administering the FMS

An initial install 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 has been installed, you can start/stop an FSC so that you can add volumes. For additional information, see [Introduction to administering FSCs](#).

You can also customize your FMS configuration after the initial installation. For sample configuration examples, see [About the sample FMS configurations](#).

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.

Note Windows 7 or later is required for full UTF-8 support. Windows XP and earlier versions are not supported by this UTF-8 implementation.

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. For more information about using this utility, see the [Utilities Reference](#).

There are circumstances when you will need to manage your FMS file server cache(s). For example, when you make a change to the `fmsmaster.xml` file on the master host, you must restart all FSCs. To manually change your FMS configuration, you might need to install and/or uninstall components.

Note In the following instructions, `FMS_HOME` refers to the location the FSC is installed. This may or may not be the same setting as the `FMS_HOME` system environment variable, set in the user's environment.

Managing your FSC on Windows

Following are tools to assist you in administering your FMS configuration running on Windows:

- To run the FSC as a service on Windows, install the FSC using the **installfsc** batch script by completing the following steps:
 1. Ensure **JAVA_HOME** and **FMS_HOME** are set to the correct paths.
 2. Run **installfsc %JAVA_HOME% %FMS_HOME% fscid**

If you installed the FSC as a service, you must start the service via Windows Services. Manage your FSC service via the **Windows Services** dialog box (**Start→Control Panel→Administrative Tools→Services**).

- To verify the FSC is running, enter the following:

```
%FMS_HOME%\fscadmin -s http://hostname:port  
FSC_hostname_user/status
```

Example: **%FMS_HOME%\fscadmin -s http://evalwin8:3805 dev15/status**

- To check the status of the read and write cache, enter the following:

```
%FMS_HOME%\fscadmin -s http://hostname:port  
FSC_hostname_user/cachesummary/read
```

Example: **%FMS_HOME%\fscadmin -s http://evalwin8:3805
dev15/cachesummary/read**

```
%FMS_HOME%\fscadmin -s http://hostname:port  
FSC_hostname_user/cachesummary/write
```

Example: **%FMS_HOME%\fscadmin -s http://evalwin8:3805
dev15/cachesummary/write**

- To stop the FSC, enter the following:

```
%FMS_HOME%\fscadmin -s http://hostname:port  
FSC_hostname_user/stop
```

Example: **%FMS_HOME%\fscadmin -s http://evalwin8:3805
dev15/cachesummary/stop**

Managing your FSC on UNIX

Following are tools to assist you in administering your FMS configuration running on UNIX:

- To verify the FSC is running, enter the following:

```
./fscadmin.sh -s http://hostname:port fsc_hostname_user/status
```

Example: **./fscadmin.sh -s http://evalsun8:3805 dev15/status**

- To check the status of the read and write cache, enter the following:

```
./fscadmin.sh -s http://hostname:port  
fsc_hostname_user/cachesummary/read
```

Example: **%./fscadmin.sh -s http://evalsun8:3805 dev15/cachesummary/read**

```
%./fscadmin.sh -s http://hostname:port
fsc_hostname_user/cachesummary/write
```

Example: **%./fscadmin.sh -s http://evalsun8:3805 dev15/cachesummary/write**

- To stop the FSC, enter the following:

```
%./fscadmin.sh -s http://hostname:port fsc_hostname_user/stop
```

Example: **%./fscadmin.sh -s http://evalsun8:3805 dev15/stop**

The FSC can fail to start after a reboot of UNIX 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 PLM 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.

Manually configuring an FSC

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 following values into the appropriate location within the **fmsmaster.xml** file.

fmseenterprise id=	<i>enterpriseID</i>
volume id=	<i>volumeUid</i>
root=	<i>wntPath</i>
root=	<i>unixPath</i>

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. For information on stopping and starting FSCs, see [Managing your FSC on Windows](#) and [Managing your FSC on UNIX](#).

Maintaining the FSC whole file cache

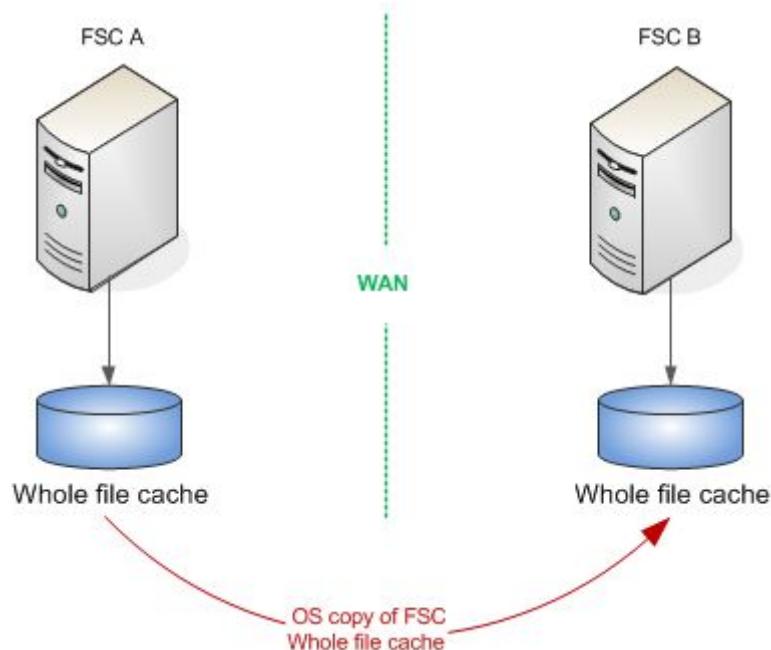
Run the **FSCWholeFileCacheUtil** utility, stored in the *FSC_HOME/bin* directory to purge files from the whole file cache. Purge the cache based on file age, subdirectory size, and/or available disk space.

This utility also purges misplaced files.

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 multisite configuration between the two enterprises.
- Replicate the **ImanFile** objects belonging to the owning site as stubs at the remote site.

For more information, see the *Multi-Site Collaboration Guide*.

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).

Benefits of whole file cache (WFC)

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.)

For IT data storage deployments:

- 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.
- 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.

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.

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 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.

- 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.

- WebRAID considerations

To use WebRAID with a forward proxy, the forward proxy host and port for the appropriate protocols (HTTP or HTTPS), must be specified in the **FMS_HOME\fcc.properties** file as follows:

```
http.proxyHost=forward-proxy-host  
http.proxyPort=forward-proxy-server-port  
https.proxyHost=forward-proxy-host  
https.proxyPort=forward-proxy-server-port
```

You must also configure the same information in the **fwdproxy_cfg.properties** file in the Teamcenter client communication system (TCCS) configuration directory.

- 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.

For more information, see [Shutting down a TCCS/FCC instance](#).

- 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.

For more information, see [Administering TCCS and its container applications](#).

For more information, see [Restarting an FCC](#).

- 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.

For more information, see [Reconfiguring an FCC](#).

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.

For more information about when and why to restart a TCCS/FCC instance, see [Elements requiring a restart of an FCC](#).

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 PLM 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 Vista and later (including Windows 7), 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 PLM 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.

For more information about running the **fccstat -kill** command, see method 2 that follows.

For more information about working with Windows shutdown scripts, see the Microsoft documentation at:

<http://technet.microsoft.com/en-us/library/cc753404.aspx>

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.

For more information, see [Reset a user's environment](#).

- Method 1
 - o 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 nonidle 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 nonidle 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

- o Run **\$FMS_HOME/bin/fccstat -kill**.

This method stops a TCCS/FCC if it is idle or nonidle, 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 **tspstat** or **tmemstat** utility to stop the shared TCCS process.

- Method 4 (only if TCCS/FCC running in foreground)

- o 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.)

- o 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 (UNIX only)

- o Run the operating system **kill** command without the **-9** option.

This method stops an FCC if it is idle or nonidle, 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 (UNIX only)

- o 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)

- o 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

Restart an FCC

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 master configuration file (**fmsmaster.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.
For more information, see [Shutting down a TCCS/FCC instance](#).
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.

For more information on administering TCCS container processes, see [Administering TCCS and its container applications](#).

Elements requiring a restart of an FCC

- 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
FCC_CachePurgeCycle

- 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.

For more information, see [Shutting down a TCCS/FCC instance](#).

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 after performing Step 3 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. Corruption in any of these files can prevent the FCC from restarting:

- **fcc.lck**
- **fms.hsh**

- **fms.mf**
 - **fms.set**
 - 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.xml** file and the master 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 master configuration file (**fmsmaster.xml**).

For example, the FCC detects when changes are made to the following **FCCDefault** parameters in the master configuration file and reconfigures itself accordingly, without interrupting FCC client application service:

- o **FCC_TransientFileFSCSource**
- o **FCC_EnableDirectFSCRouting**
- o **FCC_WebRaidThreshold**
- o **FCC_MaxWANSources**
- o **FCC_FSCConnectionRetryInterval**

The FCC also detects when changes are made to the following FCC configuration elements in the master configuration file and reconfigures itself accordingly, without interrupting FCC client application service:

- o **FCCDefaultssite**
- o **parentfsc**
- o **assignment**
- o **assignedfsc** elements within the **clientmap** element
- o **directfscroute** elements generated by the **parentfsc** file from resource elements in the master configuration file

- o **directtransientvolume** elements generated by the **parentfsc** file from resource elements in the master 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.

- o 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.xml** file or the **fcc.xml** file. Clients receive changes to immutable parameters when their FCCs are restarted.

For more information, see [Restarting an FCC](#).

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.

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.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 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.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"/>
```

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.

You can import the audit log information into standard text or word processors for correlation and examination.

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.

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 nonticketed 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. Nonticketed 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 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 **log4j** 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.

For information about defining formats and using log property elements, see [Format specifications](#) and [Audit log properties](#).

3. Enable the audit loggers using **fscadmin** commands, for example:

```
fscadmin -s http://myserver:4445
```

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. Modify the **log4j.xml** file to permanently set the audit logger level to **info** and tune the **log4j** configuration if required.

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** 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 nonticketed 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),
        PrimaryTrānsactionID, 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),
        PrimaryTrānsactionID, 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),
        PrimaryTrānsactionID, Operation, RequestMethod, RequestRemoteAddr,
        RequestHeader(User-Agent), RequestLine" overridable="false"/>
    <property name="FSC_AuditLogWebOperationStopFormat" value="Text(webopstop),
        PrimaryTrānsactionID, StatusCode, Message, ResponseHeader(Content-Encoding),
        TargetBytes, ActualBytes, ResponseStreamStatus, Deltams"
        overridable="false"/>
</fscdefaults>
```

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.
For more information about transaction IDs, see [Transaction identifiers \(IDs\)](#).
- 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.
TicketFilestoreIDs 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 .
<ul style="list-style-type: none">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.
<ul style="list-style-type: none">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 .
<ul style="list-style-type: none">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 - 2012/01/26-07:54:51,365 UTC - myhost123 - Active audit entry formats:
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(request)|,|PrimaryTransactionID|
,|RequestRemoteAddr|,|RequestHeader(X-Route)|,|RequestHeader(User-Agent)|,|RequestLine|,
|RequestHeader(Range)|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(priopstart)|,|PrimaryTransactio
nID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,|RequestHeader(X-Route)|,|RequestHea
der(User-Agent)|,|RequestHeader(Range)|,|TicketVersion|,|TicketAccessMethodNice|,|TicketI
sBinaryNice|,|TicketSignature|,|TicketExpiresTime|,|TicketUserID|,|TicketSiteID|,|TicketG
UID|,|TicketFilestoreIDs|,|TicketRelativePath|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(priopstop)|,|PrimaryTransactio
nID|,|StatusCode|,|Message|,|ResponseHeader(Content-Encoding)|,|TargetBytes|,|ActualBytes|
,|ResponseStreamStatus|,|DeltaMS|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(subopstart)|,|TransactionID|,|O
peration|,|TicketVersion|,|TicketAccessMethodNice|,|TicketIsBinaryNice|,|TicketSignature|
,|TicketExpiresTime|,|TicketUserID|,|TicketSiteID|,|TicketGUID|,|TicketFilestoreIDs|,|Tic
ketRelativePath|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(subopstop)|,|TransactionID|,|St
atusCode|,|Message|,|DeltaMS|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(webopstart)|,|PrimaryTransactio
nID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,|RequestHeader(User-Agent)|,|Request
Line|,|
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(webopstop)|,|PrimaryTransactio
nID|,|StatusCode|,|Message|,|ResponseHeader(Content-Encoding)|,|TargetBytes|,|ActualBytes|
,|ResponseStreamStatus|,|DeltaMS|,|
```

The following is sample audit log output based on the previous configuration:

```
INFO - 2012/01/26-07:54:51,379 UTC - myhost123 - |,|request|,|(-7316198962075068416)fsc
_s6|,|127.0.0.1|,|null|,|FMS-FSCJavaClientProxy/8.2 (bd:20120119)|,|GET /mapClientIPtoFS
Cs?client= HTTP/1.1|,|null|,|
INFO - 2012/01/26-07:54:51,380 UTC - myhost123 - |,|webopstart|,|(-7316198962075068416)
fsc_s6|,|BootstrapHandler|,|GET|,|127.0.0.1|,|FMS-FSCJavaClientProxy/8.2 (bd:20120119)|,
|GET /mapClientIPtoFS?client= HTTP/1.1|,|
INFO - 2012/01/26-07:54:51,381 UTC - myhost123 - |,|webopstop|,|(-7316198962075068416)f
sc_s6|,|200|,|OK|,|null|,|57|,|57|,|COMPLETE|,|1|,|
INFO - 2012/01/26-07:54:51,384 UTC - myhost123 - |,|request|,|(-7316198962075068415)fsc
_s6|,|127.0.0.1|,|null|,|FMS-FSCAdmin/8.2 (bd:20120125) Java/1.5.0_11|,|GET / HTTP/1.1|,
|null|,|
INFO - 2012/01/26-07:54:51,385 UTC - myhost123 - |,|priopstart|,|(-7316198962075068415)
fsc_s6|,|CacheCommands$ClearCommand|,|GET|,|127.0.0.1|,|null|,|FMS-FSCAdmin/8.2 (bd:2012
0125) Java/1.5.0_11|,|null|,|v100|,|ADMINREAD|,|BINARY|,|739388a12ef48c3473e19bd78049661
6b989cf3b8bab1f5d5df0bb2a7d71dbl|,|2012/01/26 07:56:51|,|FSCAdmin|,|,|,|noguid
|,|[]|,|./clearcache|,|
INFO - 2012/01/26-07:54:51,388 UTC - myhost123 - |,|priopstop|,|(-7316198962075068415)f
sc_s6|,|200|,|OK|,|null|,|17|,|17|,|COMPLETE|,|3|,|
INFO - 2012/01/26-07:55:14,180 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:20120125) Java/1.5.0_11|,|GET /tc/fms/fms.teamcenter.com/g2/fsc_s6 HTTP/1.1|,|n
ull|,|
INFO - 2012/01/26-07:55:14,180 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:20120125) Java/1.5.0_11|,|null|,|v100|,|ADMIN
READ|,|BINARY|,|ca1247695734bb33ee6e65ba0fdb087587214de0b43d8da2c2eb8353a3d92e89|,|2012/
01/26 07:57:11|,|nouser|,|,|,|,|fsc_s6/config/volum
```

```
estate/nvargs/action=get;enterpriseid=fms.teamcenter.com|,|
INFO - 2012/01/26-07:55:14,181 UTC - myhost123 - |,|priopstop|,|(-362480191128027786)fs
c_s7[1]>fsc_s6|,|200|,|OK|,|null|,|6|,|6|,|COMPLETE|,|1|,|
```

The following shows an example 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 - 2012/01/26-07:54:51,365 UTC - myhost123 - Active audit entry formats:
INFO - 2012/01/26-07:54:51,378 UTC - myhost123 - |,|Text(priopstart)|,|PrimaryTransactio
nID|,|Operation|,|RequestMethod|,|RequestRemoteAddr|,| TicketAccessMethod|,|TicketIsBinar
y|,|TicketSignature|,|TicketExpiresTime|,|TicketFileName|,| TicketUserID|,|TicketSiteID|,
|TicketGUID|,|TicketFilestoreIDs|,|TicketRelativePath|,|
  ↓
-----
INFO - 2012/01/26-07:54:51,385 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|,|g1_eng_1|,|,|,|noguid
|,|[]|,|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.

Transaction identifiers (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 though 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_infodba
FSC_spandfms_infodba
FSC_flodup_infodba
FSC_yagmlsp_infodba
FSC_wndisxk_infodba

1. The first FSC (**FSC_emdbangfms_infodba**) generates a new ID.

(342342349932) **FSC_emdbangfms_infodba**

It then performs a subordinate operation (subop), appends [1] to the transaction ID, and sends the request.

(342342349932) **FSC_emdbangfms_infodba[1]**

2. The next FSC (**FSC_spandfms_infodba**) receives and joins the existing transaction ID.

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba**

It then performs a subop.

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]**

3. A third FSC (**FSC_flodup_infodba**) receives and joins the existing transaction ID.

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]>FSC_flodup_infodba**

It then performs a subop.

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]>FSC_flodup_infodba[1]**

And another subop.

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]>FSC_flodup_infodba[2]**

4. The forth FSC (**FSC_yagmlsp_infodba**) received the prior subop ([1]).

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]>FSC_flodup_infodba[1]>FSC_yagmlsp_infodba**

5. A fifth FSC (**FSC_wndisxk_infodba**) received the prior subop ([2]).

(342342349932) **FSC_emdbangfms_infodba[1]>FSC_spandfms_infodba[1]>FSC_flodup_infodba[2]>FSC_wndisxk_infodba**

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

Managing FMS host names on IBM AIX systems

IBM AIX systems cannot reliably host domain names. You must specify host names as IP addresses when any of your file server cache (FSC) servers or file client cache (FCC) clients are hosted on an IBM AIX machine.

Using host names rather than IP addresses can result in an file client cache (FCC) failure. The failure appears as a Java exception in the FCC log file. For example:

```
java.net.UnknownHostException: (hostname)
```

You must specify host names as IP addresses in the following FMS XML configuration files:

1. In the **fmsmaster.xml** file on the master FSC server:

Change all network domain names of all IBM AIX-hosted FSC servers to IP addresses. Only IBM AIX-hosted FSC servers require this notation. For example, change

```
<fsc id="myfsc" address="myAIXserver.mydomain.com:4444">
```

to:

```
<fsc id="myfsc" address="250.142.16.3:4444">
```

and change:

```
<fscimport fscid="myotherAIXfsc"  
fscaddress="myotherserver.anotherdomain.net:6666">
```

to:

```
<fscimport fscid="myotherAIXfsc" fscaddress="125.71.8.1:6666">
```

Substitute the correct IP address values as appropriate.

Fields that require this modification include:

- The **address** field of all FSC declarations.
- The **host** field of all additional FSC connection declarations.
- The **fscaddress** field of all **fscimport** declarations.
- All **address** fields of all routing declarations.
- Any other fields which contain a URI, URL, or host name.

These changes are picked up by the **clientmap** section of the FMS master configuration file using the FSC ID.

2. In the **fcc.xml** file (or equivalent) for each IBM AIX client workstation, change all FSC server network domain names to IP addresses.

For example, change:

```
<parentfsc address="myAIXserver.mydomain.com:4444" priority="0"/>
```

to:

```
<parentfsc address="250.142.16.3:4444" priority="0"/>
```

Substitute the correct IP address values as appropriate. This includes the address field of all **parentfsc** declarations.

3. If FSCs support HTTPS connections:

- a. Issue each HTTPS-supporting FSC a certificate containing its IP address (in place of its domain name).

- b. Install the new certificate and/or remove the domain name certificate. This allows the clients to validate the FSC's host name in decimal-dot notation form.
- c. Install this new certificate in the trusted certificate store of each peer FSC and client, unless they are configured to accept self-signed certificates.

Configuring FMS to run multiple versions of Teamcenter

Different versions of Teamcenter work with different versions of Java. For example, Teamcenter engineering process management 2005 SR1 works with Java 1.4, Teamcenter 8.2 works with Java 1.5, and so on.

If you are running multiple versions of Teamcenter on your system and they work with different versions of Java, you must configure your FMS client caches (FCCs) to use the Java run-time environments (JREs) with which they were installed.

1. Open the **FMS_HOME/startfcc.sh** (UNIX systems) file or the **FMS_HOME\startfcc.bat** (Windows systems) file in a plain text editor.
2. Set the **FCC_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 Siemens PLM Software Certification Database:

<http://support.industrysoftware.automation.siemens.com/certification/teamcenter.shtml>

To use the certification database, choose the platform and products you use, and then click **Show Certifications**.

Configuring multiuser support

Enabling multiuser support configures the pipe name to work with user IDs, ensuring that each user connects with a unique pipe to the appropriate FCC.

- Configuring multiuser support on Windows

On Windows systems, the pipe connection to the FMS client cache (FCC) uses a single value by default, allowing one user to connect to the FCC per machine. If multiple users run instances of Teamcenter on the same Windows machine, each user accesses the same pipe name and possibly connects to the same FCC.

- Configure multiuser support without NX

If you are running FMS 1.3 (20070409 or later) and are not running NX, the following steps configure FMS running on a Windows system to accept multiple users.

Note Local administration privileges are required to set a global environment variable.

1. In the system environment (**System Properties**→**Advanced**→**Environment Variables**→**System Variables**), set the **FMS_WINDOWS_MULTIUSER** environment variable to **true**.
2. Reboot the system to propagate this setting to all users and applications.

- Configure multiuser support with NX

If you are running a version of FMS earlier than 1.3 or are running NX, the following steps configure FMS running on a Windows to accept multiple users.

Note Local administration privileges are required to set a global environment variable.

Write permission for **%FMS_HOME%\fcc.xml** and all FMS application startup scripts is required.

1. In the system environment (**System Properties**→**Advanced**→**Environment Variables**→**System Variables**), set the **FMS_WINDOWS_MULTIUSER** environment variable to **false**.

2. Specify a user-specific pipe name for use in step 3.

For example, for user **JohnDoe**, specify
\.\pipe\FMSClientJohnDoePipe.

For an administrative user, specify
\.\pipe\FMSClientAdministratorPipe.

3. Change the pipe name entry in the **%FMS_HOME%\fcc.xml** file. The name must end with an alpha character. Pipe names ending in numeral characters (0–9) are not supported. For example:

```
<property name="FCC_ProxyPipeName"  
value="\\.\pipe\FMSClient_{$USERPipe} /tmp/FMSClientPipe" overridable="true"/>
```

\$USER is resolved by the FCC at run time to the name of the user.

4. Run the application launch script for each FMS application. This sets the pipe name in the environment for each application. For example:

```
set FCC_PROXYPIPENAME=\\.\pipe\FMSClient_%USERNAME%Pipe
```

After the user name is substituted, the pipe name must exactly match the **FCC_ProxyPipeName** value specified in the **%FMS_HOME%\fcc.xml** file.

Note The pipe name must be set dynamically at run time (not in the system environment) to properly resolve the user name (%**USERNAME**) value.

5. Reboot the system to propagate this setting to all users and applications.

Configuring FMS for HTTPS

Introduction to configuring FMS for HTTPS

You can configure File Management System (FMS) to use either the hypertext transfer (HTTP) protocol or a secure server layer (HTTPS) protocol. You 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. This protection method uses verified and trusted server certificates, private keys (your keystore), and public keys (the certificate signing request).

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_BootStrapUrls** preference. Your only post installation tasks are generating a keystore and key entry, and generating a certificate signing request.

If you select HTTP as your protocol during Teamcenter installation, but sometime *after* installation you must configure FMS to use HTTPS, you must:

- Use a trusted certificate to generate a keystore and key entry.
- Generate a certificate signing request (CSR).
- Modify the FMS master file to reflect the new HTTPS addresses.
- Add the URL of the local HTTPS host to the list of servers defined in the **Fms_BootStrapUrls** preference.
- Update any installed FCCs.

The protection inherent in HTTPS is based on a major certificate authority guaranteeing that your Web server is the entity it claims. This is accomplished by your site providing a valid certificate indicating it was signed by a trusted authority. To work with a trusted authority you must:

- Create a key pair, keeping the private key secret.
- Generate a CSR.

Trusted certificates are purchased from third-party certificate vendors, such as VeriSign or Thawte.

Note 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 PLM Software does not recommend using self-signed certificates.

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_BootStrapUrls** preference.

The only post installation tasks required to implement HTTPS is generating a keystore and key entry and generating a certificate signing request.

For more information about these tasks, see [Generate a keystore and key entry](#) and [Create a certificate signing request \(CSR\)](#).

If you chose the HTTP protocol for FMS during installation and now want to use the HTTPS protocol, you must:

- Modify the FMS master file to reflect the new HTTPS addresses.

Update the **fsc address** in the FMS master file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fmeworld SYSTEM "fmmsmasterconfig.dtd">
<fmeworld>
  <fmenterprise id="471539747">
    <fscgroup id="mygroup">
      <fsc id="FSC_myhost_userid"
          address="https://myhost.mycompany.com:4545">
        <volume id="13974756d871c1b2023">
          root="/mnt/disk1/tcapps/tceng2005sr1mp5/TC_VOL/volume1"/>
          <transientvolume id="ce8399515fead2dee4c3e79b955d8ba">
            root="/tmp/transientVolume_tceng2005sr1mp5_userid"/>
        </fsc>
        <clientmap subnet="127.0.0.1" mask="0.0.0.0">
          <assignedfsc fscid="FSC_myhost_userid"/>
        </clientmap>
      </fscgroup>
    </fmenterprise>
  </fmeworld>
```

- Add the URL of the local HTTPS host to the list of servers defined in the **Fms_BootStrapUrls** preference.

For more information about using this preference, see the *Preferences and Environment Variables Reference*.

- 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.

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 PLM 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.
	<p>Note 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.</p>
<i>keystore.password</i>	Specifies the keystore password. The password is required to open or manage the keystore.
FSC_myhost	The standard entry is changeit .
	Specifies an alias name for the certificate.
FSC_myhost.password	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.csr	Specifies the certificate (alias) password required to retrieve the certificate.
FSC_myhost.cer	Specifies the name of the certificate signing request (CSR) file. The file contains the CSR information and is sent to the signing authority.
	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

- 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
```

- 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
```

3. Run the following command to confirm that you can read the file and view the key entry:

```
>keytool -list -keystore keystore
```

4. 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
```

5. 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-----
MIIBTjCCAR8CAQAwdjELMAkGA1UEBhMCbXkxEAOBgNvBAgTB215c3RhdGUxDzANBgNVBAcTBm15
Y210eTESMBAgA1UEChMbX1jb21wYW55MRQIwEAYDVQLEw1teWNvbXBhbnkxHDAABgNVBAmtTE215
aG9zdC5teWRvbWFpb15jb20wg28WDQYJKoZIhvCNQEBBQADgY0AMIGJAoGBAJ0h3iF8KBEN2UkW
hw1dw+RlxGwcspLA3EI+6rAKa32dg/4FY89zBcUG02413X0BxQWcsRznyWFDJHLK4En7I2xeJNs
ORwfjBeF9yWd4lzaW6LATFr5T3DHaffFmSRNP1+739mpGuOr44AXBQWqZoMhecc+n/ErekM1Z
dgWTAgMBAAGADANBgkqhkiG9wOBAQQFAAOBgQCQJTqujL7GIXz0is0fUoAxtCydMiX1BeVHU+1/
IqcTh4BX8V3vJmm+kHwwKn3yeihWJzYmDdNh/uaKxO7txyFdPPDd1bdlosFc4XIZwys0jFKwGqf
MUjB9wgaKgHSRQTTcOBPEO/C1ljm8ocFNQBWySVEvAZQAmEMp90BxBt/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
-file FSC_myhost.csr
```

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:69:79:16:66:90:02:1B: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 native FSC client proxy in TcServer

The native implementation uses cURL and OpenSSL and requires a compatible trusted certificate file.

Note 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.

If your certificate authority (CA) signer is well known, the certificates may be available from the Internet. Siemens PLM Software recommends you contact your internal security team.

- 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
```

- 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.

Configuring PKI authentication

Best practices

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 PLM 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 PLM 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

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. The conventions are those listed in *Best practices*.

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, 2009
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 2009 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, 2009
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 2009 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:

```
fsc> passwordtool -encrypt trusted.jceks.lp7qZF.password
fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
fsc> passwordtool -encrypt trustedadmin.5oDHfVV.password
Bpg2TLiFDni3bT4xS4kyIaLvz5TWGLQ/GPTJN2r5T3s=
fsc> passwordtool -encrypt untrusted.jceks.2TLiFD.password
J168VL0QG4bTbJpcIls57lqwc3P42GhTnXWfogeoVWs0=
```

- 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=Bpg2TLiFDni3bT4xS4kyIaLyz5TWGLQ/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=J168VL0QG4bTbJpc1s57lqwc3P42GhTnXWfogeoVWs0=
# 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=Bpg2TLiFDni3bT4xS4kyIaLyz5TWGLQ/GPTJN2r5T3s=
```

5. Modify the **fmsmaster** configuration file.

- a. Add the following elements under the last **fmserprise** 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).

```
<fmeworld>
...
<fmserprise 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 cmd="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:

```
fsc> fscadmin -s http://myfschost:port ./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:

```
fsc> fscadmin -s http://ci6w223:7168 ./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 - 2009/07/10-18:38:55,500 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:

```
fsc> fscadmin -s http://cii6w223:7168 ./filestoredetail
*** volume filestores:
*** transient volume filestores:
*** accesson filestores:
Filestore Details: testvolarh---sy2a---bHA, root: e:\workdir\FMSShare\FMSTestExplodedWar
\cr.txt, Len: 771999, Last modified: Mar 18 17:07 EST 2005, Last access: Jul 09 17:17 EDT 009
\crlf.txt, Len: 776010, Last modified: Mar 18 17:06 EST 2005, Last access: Jul 09 17:17 EDT 2009
\FMS User Doc Java.doc, Len: 403456, Last modified: Mar 12 09:23 EDT 2007,
Last access: Mar 19 14:25 EDT 2009
\index.html, Len: 18372, Last modified: Dec 10 15:35:22 EST 2008, Last access: Jul 09 09:09:55 EDT 2009
...
\testfiles - empty
\testvol - empty
Dirs: 270, Files: 9009, Bytes: 19719498212
```

- b. Use a trusted **fscadmin** command in an untrusted installation. For example:

```
fsc> 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.

Protecting the FMS encryption key

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. The conventions are those listed in *Best practices*.

Note The following procedure describes server-side use. However, if you set up Teamcenter to use encryption, the JREs on the clients require access to the keystore. For client-side use (two-tier, four-tier, Teamcenter Environment Manager, and so on), you must make the keystore available to all clients. For example, put the certificates and keys into the keystore, make the keystore available in a network location, and propagate the keystore to the clients where it is accessible to the client JREs.

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.

```
fsc> keygen 128
5706c8eebd67eb754544ab720f08d95b
```

3. Import the key. For example:

```
fsc> keygen -importseckey -keystore trusted.jceks -storepass trusted.jceks.lp7qZF.password
-alias ent123.tickets -keypass ent123.tickets.z3nYsY.password -key 5706c8eebd67eb754544ab720f08d95b
(nothing is returned if it worked)
```

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, 2009
Entry type: keyEntry

*****
Alias name: ent123.trustedadmin
Creation date: Jul 10, 2009
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 2009 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

*****
```

5. Use the **keygen** script to import the key file into the keystore. For example:

```
keygen -importseckey -keystore keystorfilename -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
```

6. Update any other keystores used by the system.
7. Create and/or modify the FSC property files.
 - a. Encrypt the keystore and key/alias password values using the **passwordtool** script. For example:

```
fsc> passwordtool -encrypt trusted.jceks.lp7qZF.password
fcLxB/oeZ+IeNnP/vofAqFpDqmJdSyaU0y+EHU0ffRc=
fsc> passwordtool -encrypt ent123.tickets.z3nYsY.password
vhTHTCLYz9BxE8TN4MpLFNIFkIrDMCfU7mh+pYbqfcw=
```

- b. Add the following properties to the **fsc.fcid.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=
```

8. In the **fmsmaster** configuration file, add the following elements under **fscadminpolicies** and before **fscdefaults**. For example:

```
<fmeworld>
  ...
  <fmseenterprise id="ent123">
    ...
    <!-- policy and fscadminpolicies elements would be here -->
    ...
    <ticket version="M050" keystorealias="ent123.tickets" />
    <ticket version="v100" keystorealias="ent123.tickets" />
    <ticket version="F100" keystorealias="ent123.tickets" />
```

The system uses the **keystorealias** attribute to retrieve the password from the properties file and access the key in the keystore.

9. 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:

```
fsc> fscadmin -s http://cii6w223:7168 ./config/reload
Initial configuration hash: efed77c0315fc5f9bf289cb4db0b164f
Configuration reload successful.
Final configuration hash: ac718b9778cbcfebcabea266b3c1155a
```

10. Restart the FSC.

The ticketing keys are applied.

11. 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+pYbqfcw=
```

12. Use the **fscadmin** command to confirm that the keys/certificates are available. For example:

```
fsc> fscadmin -s http://cii6w223:7168 ./keystoreinfo
Keystore info:
# of private keys: 1, aliases: [ent123.trustedadmin]
# of secret keys: 1, aliases: [ent123.tickets]
# of certificates: 1, aliases: [ent123.trustedadmin]
```

13. Check in the FSC log files. For example:

```
...  
INFO - 2009/07/10-18:38:55,500 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)  
...
```

14. Verify that the key has been moved 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:

```
fsc> fscadmin -s http://cii6w223:7168 ./keystoreinfo  
Error, server returned status code: 400, status message: TICKET_VALIDATION_FAIL_0
```

15. 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.

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 UNIX, run the following operating system command:

```
ntpdate -u pool.ntp.org
```

Configuring a PAC file to run the FMS Java applet

If you are running the thin client on Internet Explorer and using a proxy auto-config (PAC) file to access the Web server, the file may not correctly launch the FMS server.

If you encounter this problem, create the following two registry keys:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings]
"AutoConfigUrl"="http://host-name/PAC-file-name"
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings]
"AutoConfigUrl"="http://host-name/PAC-file-name"
```

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.xml** configuration because no FSCs are required to transport the files, and thus no **fmsmaster.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. Thus, four-tier transient file operations require transient volumes to be configured in the **fmsmaster.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 **userids**, as the **TcServer** process.

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** environment variable/preference is always overridden by a user-specific directory; the values set for any transient volume-related preferences and/or environment variables are ignored.

If **TC_TMP_DIR** is set the value used for **Transient_Volume_RootDir** is:

- **\$(TC_TMP_DIR)\${USER}2TierTransientVolume** on UNIX
- **TC_TMP_DIR%USERNAME%2TierTransientVolume** on Windows

If **TC_TMP_DIR** is not set the value used for **Transient_Volume_RootDir** is:

- **/tmp/\${USER}2TierTransientVolume** on UNIX
- **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** environment variable/preference is a multi-value variable used to provide UNIX 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** environment variable/preference specifies the node name of the transient volume. It is a location-based logical identifier, and is generally set to the hostname of the local machine by the **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.

Configuring transient volume elements in the master configuration file

For four-tier client file access, transient volumes must be declared within the **fmsmaster.xml** configuration file. They must be declared with the FSCs that host the transient volumes (these are the FSCs capable of file IO 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="f4986a4ab9b155d11dalafceb35f55ac" root="/m/v10002/transientVolume_v10002a" />  
</fsc>
```

Any change to the **fmsmaster.xml** configuration file requires all FSCs to be configured as configuration masters to have their master 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 master configuration file:

1. Run the **backup_xmlinfo** utility and examine the **backup.xml** file created by the utility.

2. Edit the FMS master 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 FSC(s) that use this configuration information, or reload the configuration using the **fscadmin** utility. For information about stopping and starting FSCs, see [Managing your FSC on Windows](#) and [Managing your FSC on UNIX](#).

Reload the configuration across the entire FMS network by running the following command:

```
fscadmin -s http://fschost:fscport ./config/reload/all
```

For more information about the **fscadmin** utility, see the [Utilities Reference](#).

Modifying 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 master configuration file is determined by a combination of the site ID, the value of the **Transient_Volume_RootDir** environment variable/preference, and the value of the **Transient_Volume_Installation_Location** environment variable/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 master configuration file.

For more information, see [Transient volume configuration components](#).

Note **tc_profilevars** sets the **Transient_Volume_Installation_Location** environment variable/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 UNIX platform.

To set the **TcServer** context on Windows and run the utility:

1. Run the Teamcenter command prompt to open a command window by clicking **Start→Program Files**.
2. In the command prompt, type **backup_xmlinfo -u=infodba -p=infodba -g=dba** where *infodba* and *dba* is 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 UNIX 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=infodba -p=infodba -g=dba** where *infodba* and *dba* is 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.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 master configuration file is determined by a combination of the site ID, the value of the **Transient_Volume_RootDir** environment variable/preference, and the value of the **Transient_Volume_Installation_Location** environment variable/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 master configuration file.

For more information, see [Transient volume configuration components](#).

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** environment variable/preference to the value used for the desired **TcServer**, then complete the steps listed in [Modifying the transient volume ID for the current server context](#).

Modifying transient volume ID components

Modifying either the site ID or the **Transient_Volume_RootDir** environment variable/preference has far reaching impact, affecting multiple transient volumes defined in the system.

Warning Siemens PLM 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** value invalidates all transient volumes on the platform (Windows/UNIX) that was modified because this environment variable/preference defines the path to the transient volumes for the specified platform.

To modify the **Transient_Volume_RootDir** environment variable/preference:

1. Change the value of the **Transient_Volume_RootDir** environment variable/preference to the new path name of the transient volume on the server.
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. For more information about using this application, see the *Organization Guide*.

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 master configuration file

For more information about using this application to manage volumes, see the *Organization Guide*.

You can also create, manage, review, purge, and move volumes using various Teamcenter utilities.

For more information about these utilities, see the *Utilities Reference*.

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.

For more information, see *Introduction to default local volumes*.

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**.

For more information about creating volumes, see the [Organization Guide](#).

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.

For more information about creating users, see the [Organization Guide](#).

You can also define a default volume for a user by choosing **Edit→User Settings** and selecting a default volume from the **Default Volume** drop-down list.

For more information about the **User Settings** dialog box, see the [Rich Client Interface Guide](#).

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.

For more information, see [Moving files in batch for default local volumes](#).

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 or the thin 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

For more information about setting default local volumes, see the *Organization Guide*.

Enabling 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 **Index** 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**.

For more information about creating volumes, see the *Organization Guide*.

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.

For more information about creating users, see the *Organization Guide*.

You can also define a default local volume for a user by choosing **Edit→User Settings** and selecting a default local volume from the **Default 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.

For more information about installing the Dispatcher Scheduler, see *Getting Started with Dispatcher (Translation Management)*.

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.

Moving files in batch for default local volumes

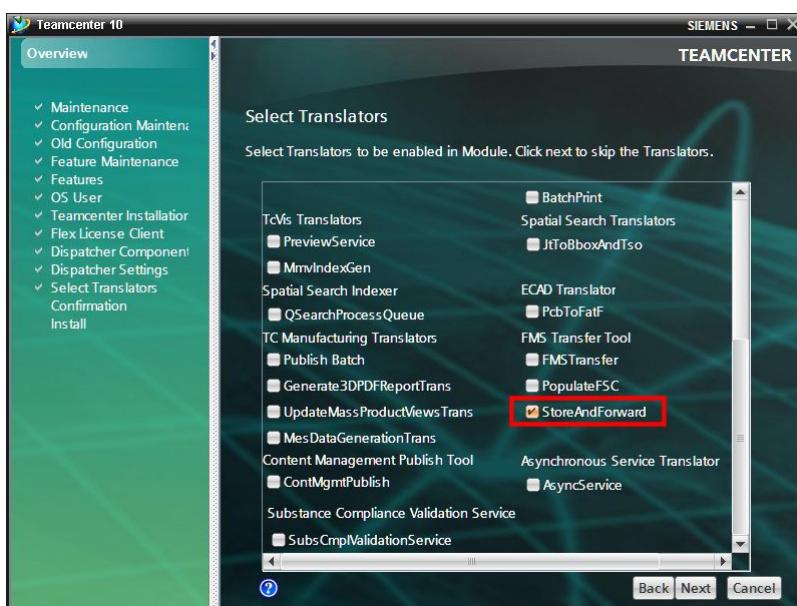
Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the final destination volume. This functionality is also known as *store and forward*.

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.
 - a. In Teamcenter Environment Manager (TEM), choose **Enterprise Knowledge Foundation® Dispatcher Server**.
 - b. 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.



Installing the StoreAndForward translator

2. Set the **TC_Store_and_Forward** preference to **true**.

You can set the preference by choosing **Edit→Options** to open the **Options** dialog box. Use the **Index** tab in this dialog box to locate the preference.

3. Set the **FMS_SAF_Batch_Transfer_Enabled** preference to **true**.

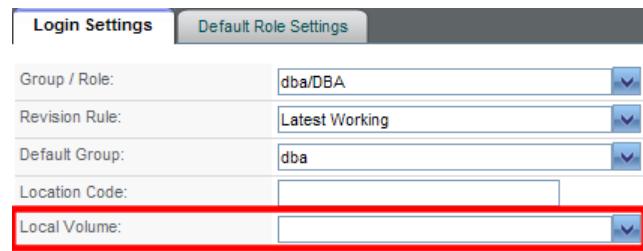
This switches the transfer mode from moving files one at a time (using the **fmstranslator** translator) to moving files in batches (using the **store_and_forward** translator).

4. Set the local volume for the user in the **User Settings** dialog box in the Organization application.

Rich client user settings



Thin client user settings



5. Optionally, view the files in local volumes to be moved to default volumes by running 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.

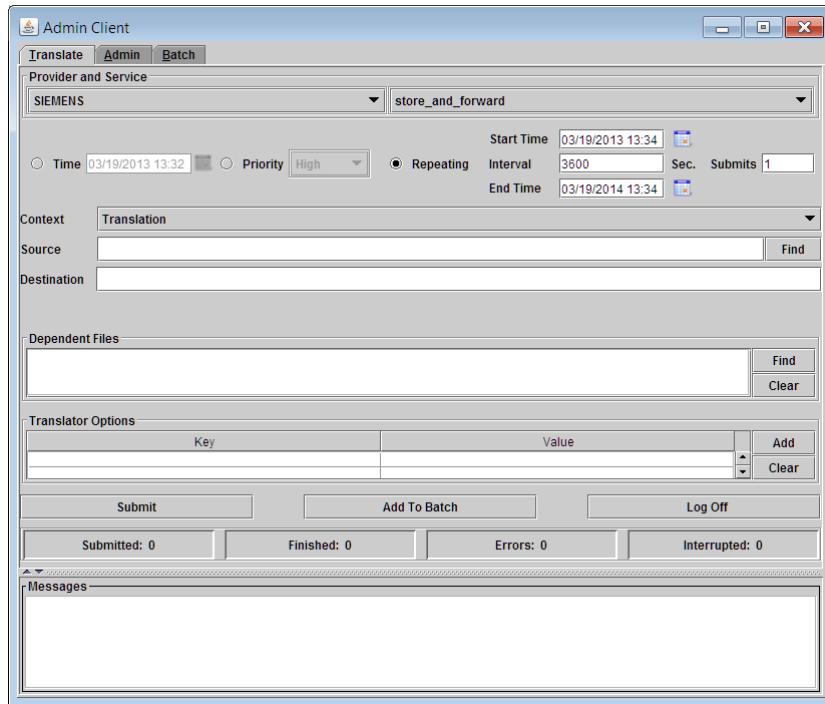
This translator in turn runs the **move_volume_files** utility with the **-transferaf** argument, which queries the database for the user files in local volumes and transfers them to their respective default volumes. It also automatically schedules a task to clean up the files from the local volumes.

For more information, see the *Dispatcher Server Translators Reference Guide*.

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.



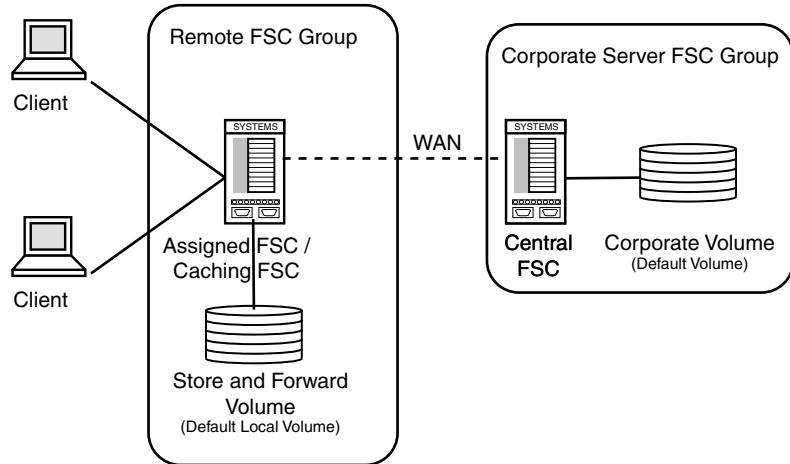
Running the **store_and_forward** translator in the Dispatcher Admin Client

7. After the translator setup is complete, verify the translator installation:
 - a. Create datasets in the local volume using the rich client or thin 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 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 FMS server cache (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.



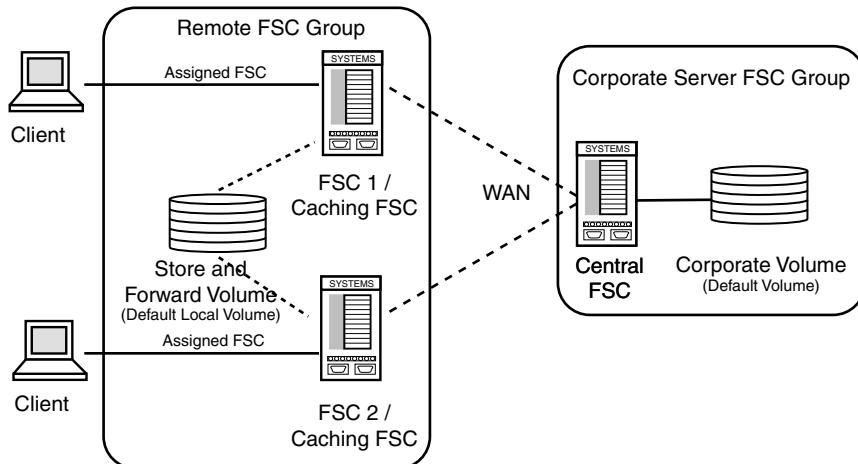
Add a default local volume to an existing caching FSC using the Organization application.

For more information about creating volumes, see the [Organization Guide](#).

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.



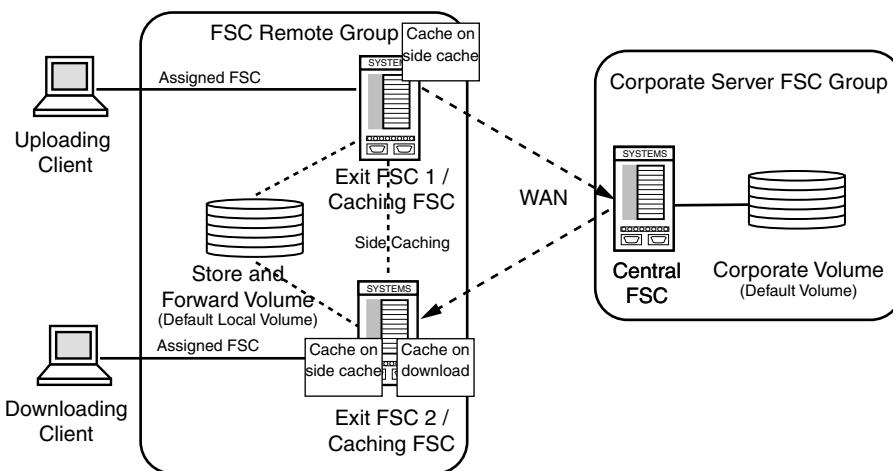
Use the **filestoregroup** element in the **fmsmaster.xml** file to add a default local volume, cross-mounted on two FSCs.

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 master configuration file (**fmsmaster.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 master 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, an 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 PLM 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_BootstrapUrls** preference or the **parentfsc** list in the **fcc.xml** configuration file.

Volume failover features are configured with the **priority=** attributes of a number of elements in the **fmsmaster.xml** file.

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.

Working with configuration failover

Configuration failover allows the client to fail over from one configuration server FSC to another, based on the **Fms_BootstrapUrls** preference and **parentfsc** configuration information.

- For FSC clients, this functionality is implemented by specifying backup configuration servers with the **Fms_BootstrapUrls** preference. The preference provides a list of bootstrap (configuration server) FSCs, from which the **assignedfsc** list is obtained. This list is ordered from highest priority to lowest priority.

Ensure that you include in the **Fms_BootstrapUrls** preference all of the FSC addresses that support the failover volumes. Also add the FSCs to the **FMS_HOME\fcc.xml** file as priority **0** so that you can get a connection and automatically pick up the server as long as one of the FSCs is up and running in the configuration.

The bootstrap servers specified in the **Fms_BootstrapUrls** preference are *not* intended to be analogous to the **assignedfsc** list in a **clientmap** entry. The FSCs in this list are intended to be analogous to the **parentfsc** elements in an **fcc.xml** configuration file.

- For FCC clients, this functionality is implemented by specifying backup **parentfsc** elements in the **fcc.xml** configuration file.

The **parentfsc** list is *not* intended to be analogous to the **assignedfsc** list in a **clientmap** entry unless **assignment mode="parentfsc"** is also specified in the **fcc.xml** configuration file.

Working with 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 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:

- o 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.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.

- o 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.
- o If the **FCC_EnableDirectFSCRouting** parameter is set to **true** (the default setting), and the **resource** element¹ is served within the same **fscgroup** element as the client's primary **assignedfsc** element, the request

1. The list of **resource** elements includes **volume**, **transientvolume**, **logvolume**, and **accession**.

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 **resource¹** and **filestoregroup** elements of the **fsc** and **loadbalancer** elements in the **fmsmaster.xml** file and delivered to the FCC with the **assignedfsc** list on FCC initialization.

- o 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).
 - o 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.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.xml** file to implement volume failover as a part of FMS message routing. These elements include:

- o The **resource¹** and **filestoregroup** elements of **fsc** and **loadbalancer** elements, which apply most directly to the selection of target resource servers for FMS routing.
- o The **entryfsc** elements of **fscgroup** elements, particularly when the **entryfsc** element refers directly to volume servers.
- o The **exitfsc** elements of **fscgroup** elements, particularly when the **exitfsc** element refers directly to volume servers.
- o The **fscimport** elements of **defaultfscimport** elements, particularly when the **fscimport** elements refer directly to volume servers in the **fmsmaster.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.xml** file information. If an FSC server has an inaccurate configuration, it is prone to routing errors, which can cause user transaction failure.

Configuring volume failover

Implement volume failover by specifying backup volume servers using the **priority=** attributes of **resource¹** and **filestoregroup** elements of the **fsc** and **loadbalancer** elements in the **fmsmaster.xml** file.

The **priority=** attributes on a number of other elements in the **fmsmaster.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 **resource¹** 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.xml** configuration file of the external **fmsenterprise** element.

Configuring FSC volume failover

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.

```
<fmeworld>
  <fmenterprise 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="//nfsserver1/dashshare/testvol" priority="0"/>
      </fsc>
      <fsc id="fsc4" address="http://localhost:5554">
        <volume id="testvol" root="//nfsserver1/dashshare/testvol" priority="1"/>
      </fsc>
    </fscgroup>
  </fmenterprise>
</fmeworld>
```

```
</fsc>
</fscgroup>
</fmenterprise>
</fmsworld>
```

Configuring 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 PLM 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.

Volume data

Volume allocation rules

You can move files from one volume to another using allocation rules. Use the [move_volume_files](#) utility 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	<usercriteria name="id" value="infodba" />
Group	<usercriteria name="group" value="dba" />
Project	<usercriteria name="project" value="FMS" />

Item criteria	Sample rules
Item ID	<itemcriteria name="id" value="ABC" />
Owning user	<itemcriteria name="user" value="infodba" />
Owning group	<itemcriteria name="group" value="dba" />
Owning project	<itemcriteria name="project" value="FMS" />
Attached dataset type	<itemcriteria name="type" value="UGMaster" />

Item revision criteria	Sample rules
Owning user	<itemrevisioncriteria name="user" value="infodba" />
Owning group	<itemrevisioncriteria name="group" value="dba" />
Owning project	<itemrevisioncriteria name="project" value="FMS" />
Attached dataset type	<itemcriteria name="type" value="UGMaster" />

Dataset criteria	Sample rules
Dataset type	<datasetcriteria name="type" value="UGMaster" />
Owning user	<datasetcriteria name="user" value="infodba" />
Owning group	<datasetcriteria name="group" value="dba" />
Owning project	<datasetcriteria name="project" value="FMS" />
Volume ID	<datasetcriteria name="volume" value="vol1" />

Volume data criteria	Sample rules
Volume free space	<volumecriteria name="availablespace" value="50GB" />

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=infodba -p=infodba -g=dba -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.

For an example, see [Sample volume allocation rules XML file](#).

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=infodba -p=infodba -g=dba -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=infodba -p=infodba -g=dba -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.

For more information about using this utility, see the [Utilities Reference](#).

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) #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="infodba" />
        <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="infodba" />
        <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">
```

```

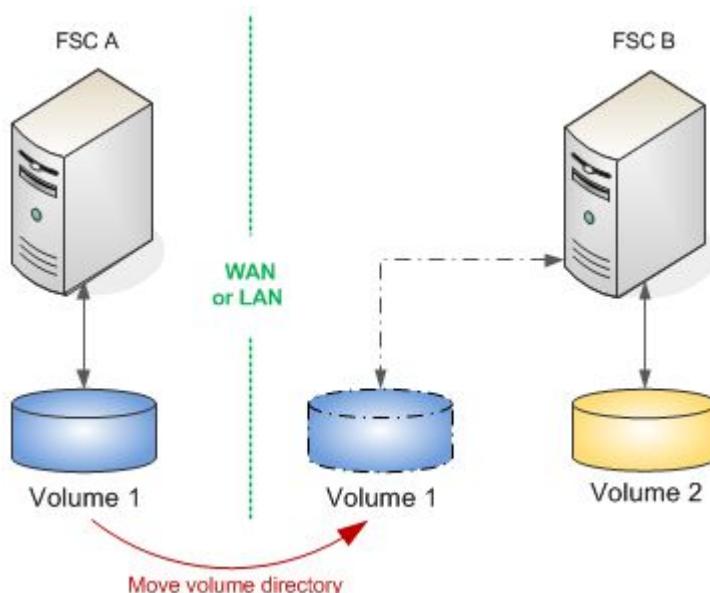
<datasetcriteria name="type" value="JT" />
<usercriteria name="project" value="CarPart" />
</reallocationrule>
</volumereallocation>

```

Moving 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=readonly -u=infodba -p=password -g=dba
```
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 *FMS_HOME*) to retrieve the FMS master configuration of **FSC A**. Make a note of 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 master configuration file for **FSC A**. For example:


```
fscadmin -s http://FSCA_address:port ./config/removevolume/nvargs/
volumeid=volume-UID;enterpriseid=site-ID;fmsconfigentryonly=1
```

5. Use the **fscadmin** utility to add the destination location and volume UID to the FMS master 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/nvargs/
root=destination-locatiōn;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/nvargs/
root=destination-locatiōn;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/nvargs/
root=destination-locatiōn;volumeid=volume-UID;
loadbalancerid=load-balancer;enterpriseid=site-ID
```

6. Using your operating system tools, delete the volume under **FSC A**.

7. Use the **backup_modes** utility to put Teamcenter in read-only mode. For example:

```
backup_modes -m=normal -u=infodbb -p=password -g=dba
```

8. In Teamcenter, using the Organization application, modify the volume's path name attribute.

In the **Move Volume** dialog box, select **No** at the prompt **Do you want to move files?**, and select **Yes** to change the volume location without moving the volume data.

For more information about modifying the path name attribute, see the *[Organization Guide](#)*.

Note To move a volume from one location to another *on the same host*, see the *[Organization Guide](#)*.

Load balancing FMS data

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
parentfsc	The fcc.xml file.
assignedfsc	Within the clientmap elements in the fmssmaster.xml file.
fscmaster	The fsc.xml file.
entryfsc	Within the fscgroup elements in the fmssmaster.xml file.

Element	Location
exitfsc	Within the fscgroup elements in the fmsmaster.xml file.
defaultfsc	Within the multisiteimport elements within the fmsenterprise elements.
volume	Within the filestoregroup and fsc elements in the fmsmaster.xml file.
transientvolume	Within the filestoregroup and fsc elements in the fmsmaster.xml file.
logvolume	Within the filestoregroup and fsc elements in the fmsmaster.xml file.
accesson	Within the filestoregroup and fsc elements in the fmsmaster.xml file.
filestore	Within the fsc elements in the fmsmaster.xml file.

Note The **routethrough** elements within **grouproute** elements of the **fmsmaster.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 master configuration file contains the following elements, the FCC directs requests for data on the volume **voll** approximately equally among FMS servers **fsc1** and **fsc2**.

Tip Legacy FCCs (configured prior to Teamcenter Engineering Process Management 2005 SR1 MP2) cannot implement software load balancing. However a partial load balancing is applied to legacy FCC clients. This is provided by the FSC, which randomizes the FSC order for the equal priorities each FCC configuration download. This effectively assigns various FCC clients randomly to the equal priority FSCs, although each legacy FCC client still uses the list in failover order. (Legacy FCCs require all priorities to be unique.)

```
<filestoregroup id="fsgroup1">
  <volume id="voll" root="/data/voll"/>
</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 master configuration 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. 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 can not 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 master configuration file (**fmsmaster.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**, **accesson**

Load balancers support health checking for their servers. Health checking is generally configured to use a particular URL within the back-end servers.

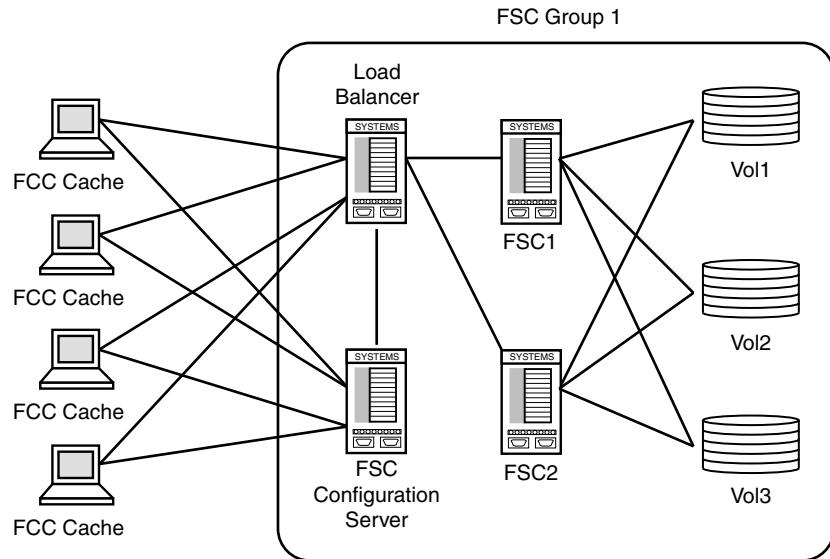
Because FSC servers support few requests that do not require tickets, many default health checks return **HTTP 400** errors that load balancers may interpret to mean service is unavailable. To avoid these errors, Siemens PLM Software recommends using **HTTP HEAD** or **GET** requests for **/Ping** or **/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. 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.

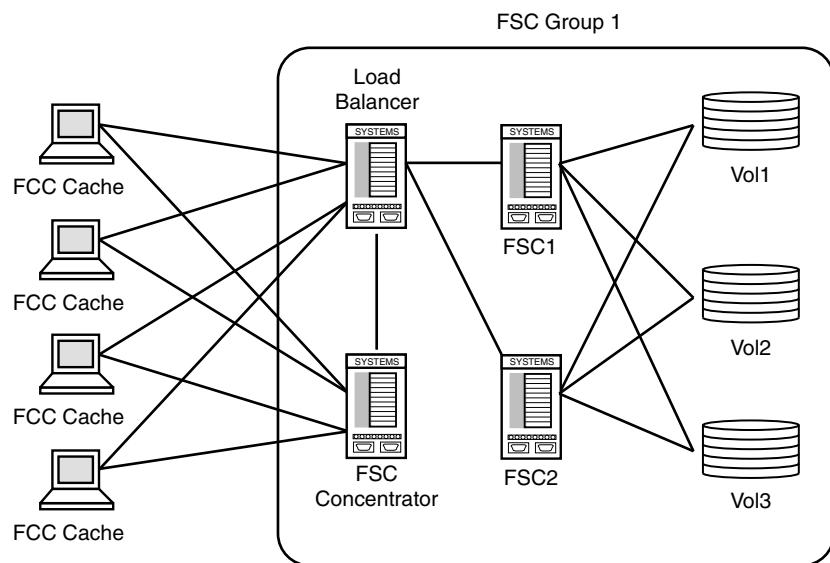
```

<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 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>

```

Working with client maps

Introduction to working with client maps

The client map element in the master configuration file (**fmsmasterconfig.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 sever 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.

Using 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 matched 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.

For more information, see [Using domain name client maps](#).

Using 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.

For more information, see [Using subnet/mask attributes in a client map](#).

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.

For more information, see [Using domain name client maps](#)

Using 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 client map element.

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>

Attribute	Use
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 FMS databases through a single FCC

Introduction to accessing multiple FMS databases through a single FCC

You can access multiple File Management System (FMS) databases through a single 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 master configuration file at the primary site to allow the FCC to determine from which FSCs to request additional site data.

Note Previous to Teamcenter 8, all methods of supporting access to multiple databases were implemented in the FMS server cache (FSC). In these situations, an FCC connected to a single parent FSC to download configuration data summarizing the client-relevant portions of the **fmsmaster.xml** file. If the master configuration file did not contain the data required to access multiple databases, the FSCs connecting to the FSC had no access to the data managed by the other database sources.

Implement this functionality by configuring the FCC's local **fcc.xml** file or the master configuration file (**fmsmaster.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.xml** file. Do this by modifying the **fcc.xml** file or **fmsmaster.xml** file to configure your FCCs to route each FMS request to a different FSC (or set of FSCs) based on the **SiteID** contained in each request ticket. Each FCC will continue to have one or more default FSC destination to which it routes requests associated with unknown **SiteIDs**.

For more information about configuring the FMS files, see example configurations in [*fmsmaster.xml configuration example*](#).

To implement this functionality:

- Each database must have a unique **SiteID**.
- 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.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.

For more information about configuring the FMS files, see example configurations in [*fmsmaster.xml configuration example*](#).

- The proper multisite FCC client configuration must be present in the **FMS_HOME** directory from which the FCC is started. Siemens PLM 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.

fmsmaster.xml configuration example

Modify your **fmsmaster.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 fsmworld SYSTEM "fmsmasterconfig.dtd">
...
<fmeworld>
  <fmenterprise id="myenterprise">
    ...
    <fccdefaults>
      <!-- All clients in this fmenterprise 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>
      </fsc>
    </fscgroup>
  </fmenterprise>
</fmeworld>
```

```

</fsc>
...
</fscgroup>
</fmsenterprise>
</fmeworld>

```

fcc.xml 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 fmssmaster.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 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 filename (**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 fmssmaster.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>

```

Compressing FMS files

Overview of compressing 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_WebRaidThreshold**) 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.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,gt,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_WebRaidThreshold** element located within the **fscdefault** element in the **fmsmaster.xml** file. Files smaller than this value are not compressed.

This value also determines the threshold file size that must be reached before WebRAID (WAN acceleration) is used. The default setting is 32 K.

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:

```
<fmeworld>
  <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" />
<property name="FSC_LogFile" value="$HOME/$FSC_ID.log|/scratch/$USER/$FSC_ID.log" 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,lna,
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" />

    <accesson id="myvolAearh---sy2a---bHA" root="$FMS_TESTFILE_DIR" username="mrd"
      textfileencoding="UTF-8" lineterminationstyle="LF" />
    <accesson id="testvolarh---sy2a---bHA" root="$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" />

</fmserprise>
</fmeworld>

```

Determining which transport method is used

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.

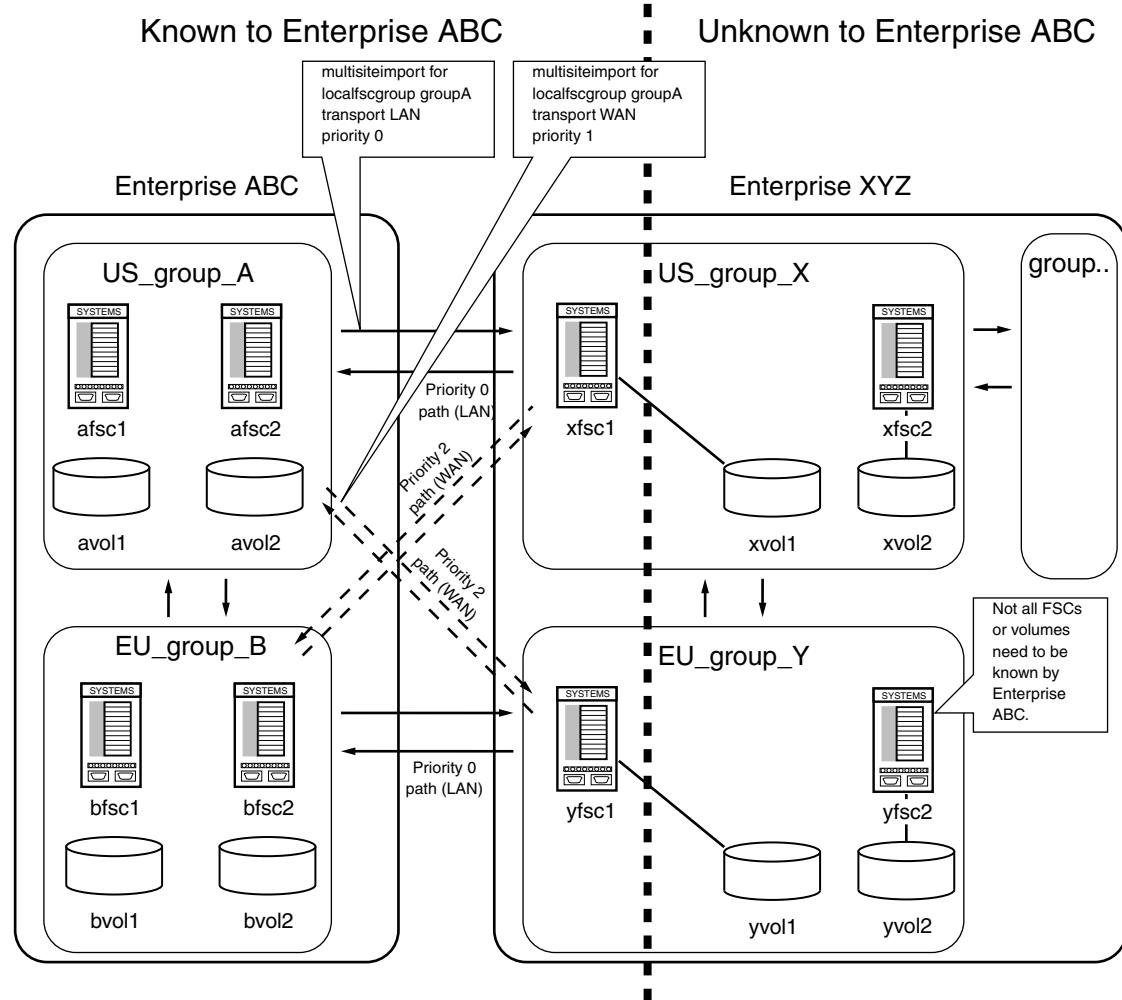
The following table indicates the transport method used to transport files, based on how you configure compression parameters, file type, and whether the file size is greater than the threshold value set for the **FSC_WebRaidThreshold** element.

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

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.)



Use the **fscgroupimport** XML element 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**.

```

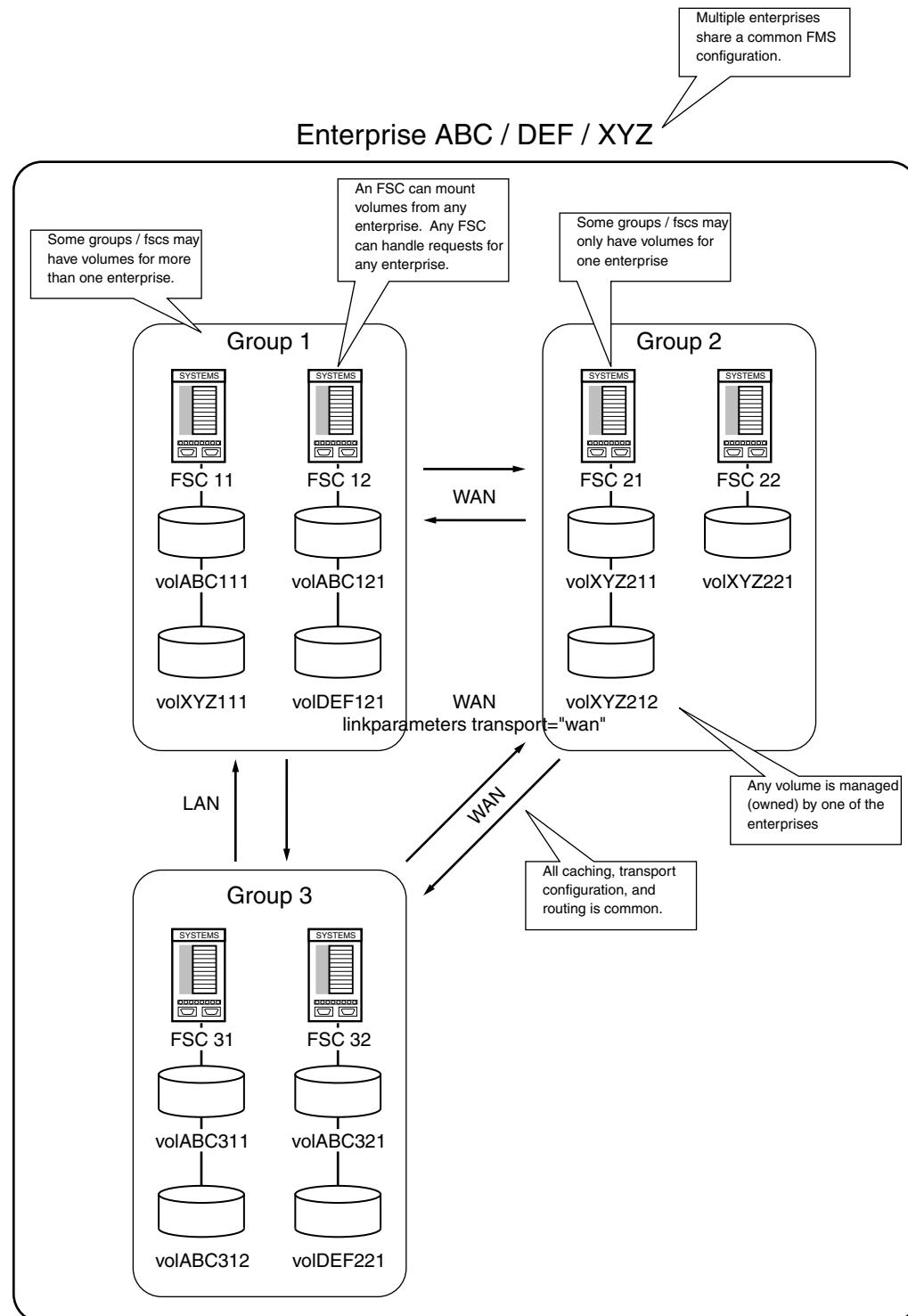
<fmeworld>
<multisiteimport siteid="XYZ">
  <defaultfscimport fscid="xfsc1" address="http://127.100.0.1:5101" priority="0"/>
  <defaultfscimport fscid="yfsc1" address="http://127.100.0.1:5102" priority="1"/>
  <fscgroupimport groupid="US_group_A">
    <!-- only address or fscid is needed not both -->
    <!-- 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>
<fmenterprise 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>
</fmeworld>
```

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 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.



In the following example code, additional sites are added to the configuration using the **fmsenterprise** element (**DEF** and **XYZ**):

```

<fmeworld>
<fmseenterprise id="ABC" >
<fmseenterprise id="DEF"/>
<fmseenterprise 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://fsc22:4544">
    <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"/>
</fmseenterprise>
</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 e-mail notification when specified metrics cross specified thresholds.

Metric	Description
Quarantined Dead Link	A link between two resources in the FSC network topology was quarantined.
All Routes Failed	All routes to resources in the FMS topology are inaccessible.
No Route Error	A client or FCC is connector to the wrong FSC server process.
Remote Admin Not Supported	The Fms_BootStrapUrls value is incorrect.
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.
Generic Error	The FSC server threw a general error.
Invalid Ticket	The FSC server encountered a invalid ticket.

Metric	Description
Expired Ticket	The FSC Server encountered a expired ticket.
Periodic Checks	The periodic FSC network, local volume, performance, and configuration checks has detected an issue.

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 screen of the J2EE 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.

Configure FMS monitoring using either:

- The **TC_ROOT/fsc/fscMonitorConfig.xml** file.
For more information about using the XML file, see [Configure monitoring with the fscMonitoringConfig.xml file](#).
- The FMS administrative interface.
For more information about using the FMS administrative interface, see [Configure monitoring with the administrative interface](#).
For more information starting the FMS administrative interface, see [Start the administrative interface](#).

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 e-mail 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 monitoring with the **fscMonitoringConfig.xml** file

1. Open the **FMS_HOME/monitoring/fscMonitoringConfig.xml** file.
2. 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 e-mail 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 e-mail responder. Multiple e-mail responders can be configured, in which case, the identifiers must be unique.

- **protocol**

Specify the e-mail protocol by which notifications are sent. The only supported protocol is SMTP.

- **hostAddress**

Specify the server host from which the e-mail 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 e-mails are sent.
 - **toAddress**
Specify the address to which the notification e-mails are sent.
 - **suppressionPeriod**
Specify the amount of time (in seconds) to suppress e-mail notification of critical events.
For more information, see the suppression period example in [*Introduction to monitoring*](#).
 - **emailFormat**
Specify the format in which the e-mail 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.
For more information, see the suppression period example in [*Introduction to monitoring*](#).
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**

Collect metric data, display results in the MBean view for this metric, and send e-mail notifications when critical events occur.

- **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 e-mail 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="svclsmtp.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" />
    <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_BootStrapUrls 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>
</MetricsConfig>
</ApplicationConfig>

```

Configure monitoring with the administrative interface

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 JVisualVVM that can be used for this purpose.

This procedures assumes you have the administrative interface running.

- 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.

- 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.

- In the same view, click the **id=EmailResponder1** value.

- (Optional) To be notified when criteria reaches the specified threshold, specify from whom, to whom, and how frequently e-mail 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 e-mails are sent.

- Host_address**

Specify the server host from which the e-mail 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 e-mail notification of critical events.

For more information, see the suppression period example in [Introduction to monitoring](#).

- To_address**

Specify the address to which the notification e-mails are sent. You can specify multiple e-mail addresses, separated by commas.

5. Click **Apply**.
6. Click **Back to Agent View**.
7. Under the **FSC_Critical_Events** heading, click **id=LoggerResponder1**.
8. (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.

For more information, see the suppression period example in *Introduction to monitoring*.

9. Click **Apply**.

Start the 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. Open the **pref_export.xml.template** file in the *FMS_HOME* directory.
2. Locate the **RunHtmlAdapter** key entry and set its value to **true**.
3. Add the following elements to the file following the **RunHtmlAdapter** setting:

```
<entry key="HtmlServerLogin" value="plmmonitor" />
<entry key="HtmlServerPassword" value="localhost" />
```
4. Save the modified file as **pref_export.xml**.
5. Open the **startfsc** file in the *FMS_HOME* directory.
6. 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
```

7. Restart the FSC and ensure there are no errors.
8. 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.

9. Log on using the default user name and password **plmmonitor** and **localhost**, respectively.

The Web application metrics appear.

10. Click any of the links for the listed metric MBean.

Improving 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_fsccache** 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=infodba -p=infodba -g=dba -item=ITEM -export_bom=yes  
-transfermode=justDatasetsOut -xml_file=tickets.plmxml
```

2. Run the **load_fsccache** utility to target the **fsc_ids** within the PLM XML file and prepopulate the cache. For example:

```
load_fsccache -u=infodba -p=infodba -g=dba -f=load -fsctargets=fsc-ID  
-plmxml=abc.xml -log_types=ALL -log_filename=c:\temp\load.out
```

Sample FMS configurations

About the sample FMS configurations

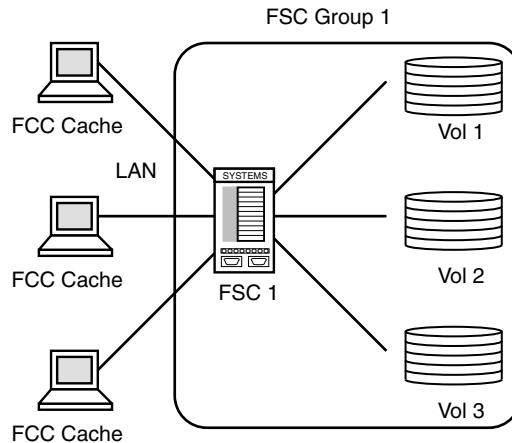
The sample Teamcenter rich client FMS configurations are fairly simple FMS deployments illustrating a single FMS capability and addressing basic FMS configuration solutions.

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 following basic two-tier Teamcenter FMS configuration:



- **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 master configuration file**

This file is served by FSC1. 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

Specifies where the FSC read and write cache files are stored. These defaults are used by any additional FSCs installed in the future.

fccdefaults

Specifies the default location of the user's cache.

fscGroup

Defines the FSC group. All FSCs must belong to one, and only one, FSC group.

fsc

Defines the FSC identifier and configures the FSC. 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

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_xml** information program to generate the volume IDs.

clientmap

Specifies that all clients on 146 prefixed network addresses are assigned to FSC1.

A sample of the master configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsconfig SYSTEM "fmsmasterconfig.dtd">
<fmeworld>
  <fmssenterprise 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>
  </fmssenterprise>
</fmeworld>
```

- **FSC configuration file**

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. Key points of this file in this configuration are:

fscmaster

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

Specifies the FSC ID for this installed FSC. This FSC ID is used to refer to the FSC definition provided in the FMS master configuration file. For example:

```
FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsconfig SYSTEM "fsconfig.dtd">
<fsconfig>
  <fscmaster serves="true"/>
  <fsc id="fsc1"/>
</fsconfig>
```

- **FCC configuration file**

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. Key points of this file in this configuration are:

parentfsc

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 master or FSC configuration files.

Assigned FSC

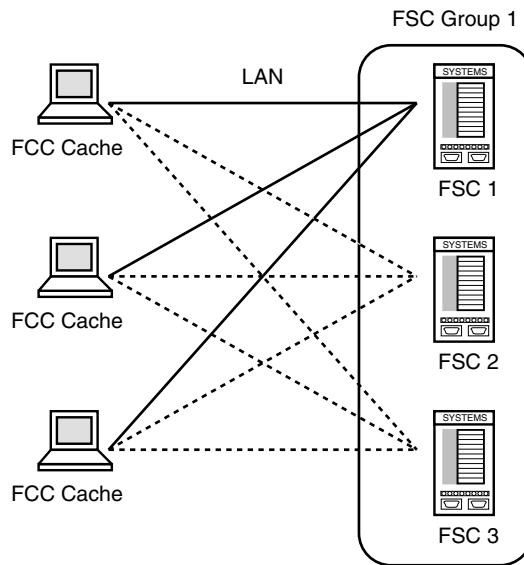
The assigned FSC is FSC1, as specified in the FMS master configuration file.

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.



- **Master configuration file**

This file is configured similar to the previous example for the single FSC configuration. 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

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 master configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmssmasterconfig.dtd">

<fmeworld>
  <fmenterprise 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>
    <clientmap subnet="146.0.0.1" mask="255.0.0.0">
        <assignedfsc fscid="fsc1"/>
    </clientmap>
</fscGroup>
</fmserprise>
</fmsworld>

```

- **FSC configuration files**

In this example, each FSC has a separate **fsc.xml** file. Key points of this file in this configuration are:

Multiple FSC configuration files

Each FSC has a separate configuration file.

fmsmaster

Specifies that FSC1 is the master configuration server in the FSC1config file. The FSC2 and FSC3 configuration files download the master 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>

```

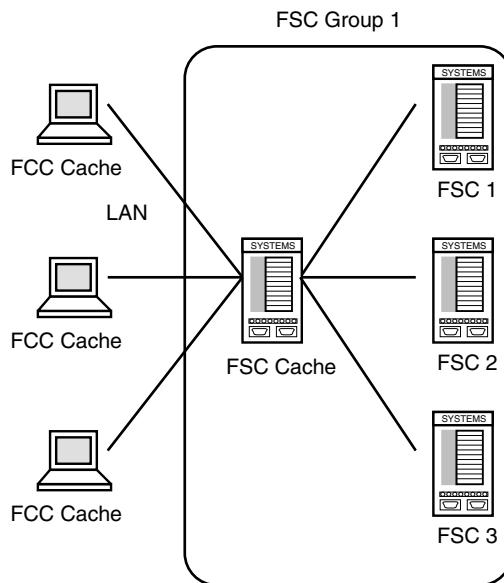
- **FCC configuration file**

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.



- **Master configuration file**

This file is configured similar to the example for the FCC Direct Connect configuration. 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

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 master configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmssmasterconfig.dtd">
<fmssworld>
    <fmssenterprise 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>
    </fmssenterprise>
</fmssworld>
```

- **FSC configuration files**

This FSC configuration file is similar to the FSC Direct Connect configuration file. Each FSC has a separate **fsc.xml** file. 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**

This example of the FCC configuration file is similar to previous examples:

```

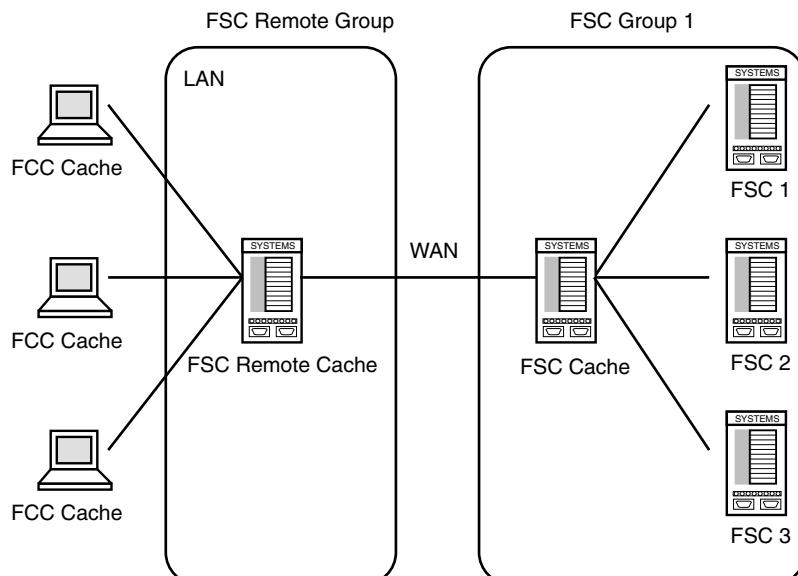
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fccconfig SYSTEM "fccconfig.dtd">
<fccconfig>
  <parentfsc address="http:// csun16.ugs.com:4444" priority="0"/>
</fccconfig>

```

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.



- **Master configuration file**

This file is configured similar to the example for the FSC cached configuration. 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

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

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 master configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fmssmasterconfig.dtd">
<fmssworld>
    <fmserprise 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>
        </fscGroup>
    </fmserprise>
</fmssworld>
```

```

<linkparameters fromgroup="fscRemoteGroup"
    togroup="fscGroup1"
    transport="wan" />
</fmsenterprise>
</fmeworld>
```

- **FSC configuration files**

This FSC configuration file for this configuration has the same key points as the FSC cached configuration.

```

FSCREMOTE CACHE
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fscconfig SYSTEM "fscconfig.dtd">
<fscconfig>
    <fscmaster serves="false" address="http://csun16.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**

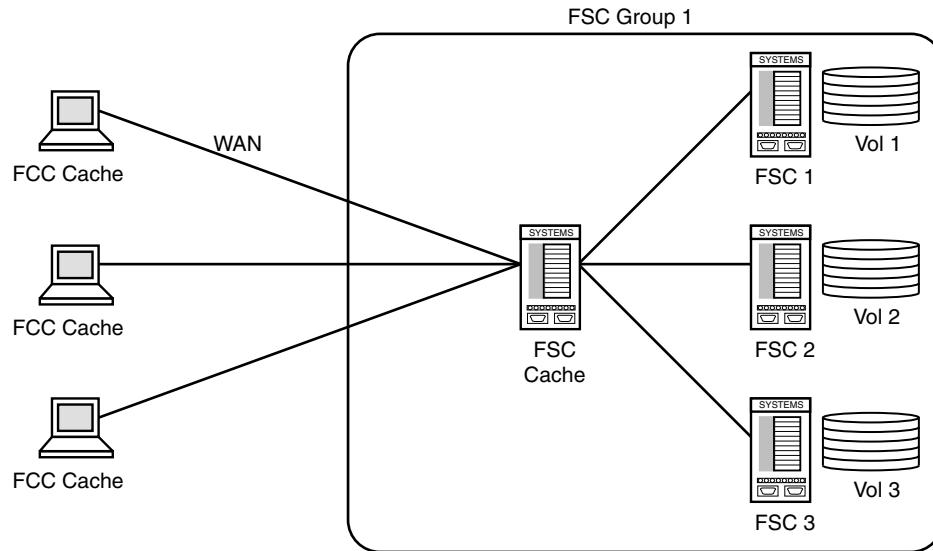
This example of the FCC configuration file is similar to previous examples:

```

<?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.



- **Master configuration file**

This file is configured similar to the example for the FSC cached configuration. 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 "fmssmasterconfig.dtd">
<fmeworld>
    <fmenterprise 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="fscl" 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>
            <clientmap subnet="146.0.0.1" mask="255.0.0.0">
                <assignedfsc fscid="fscCache" transport="wan"/>
            </clientmap>
        </fccdefaults>
    </fmenterprise>
</fmeworld>
```

```
</fmsenterprise>
</fmeworld>
```

- **FSC configuration files**

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**

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>
```

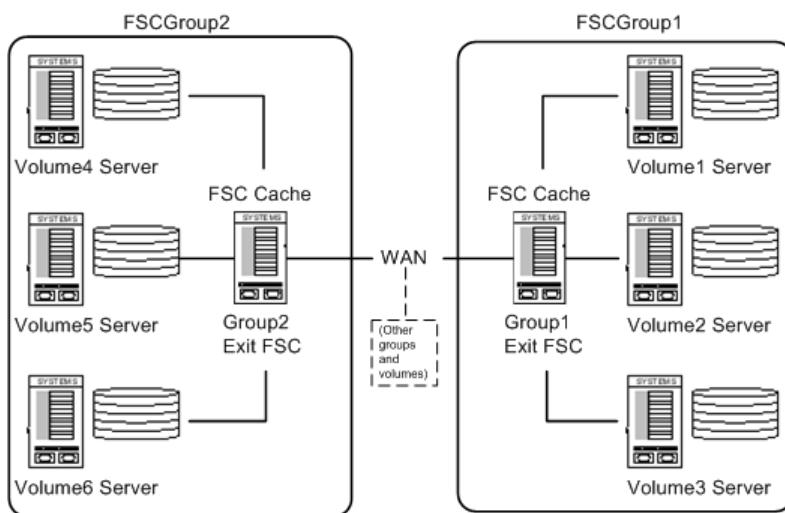
Exit FSC cache configuration

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 **CacheIfNotInLocalFSCGroup**. This prevents the exit FSC from caching group data.



- Master configuration file**

This file is configured similar to the example for the FSC cached configuration. Key points of this file in this configuration are:

ExitFSC

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

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. Thin clients always route all requests through the **AssignedFSC** element.

AssignedFSC

The **AssignedFSC** element for each group is **ExitFSC**. Therefore rich clients route requests for data served outside the group to the exit FSC first. Thin clients always route all requests through the assigned FSC.

FSC_Cache Policy

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. This setting

is particularly important for thin clients, which always route all requests through the assigned FSC.

Link parameters

WAN acceleration is enabled from the remote office to the central office using the **linkparameters fromgroup** element.

An example of the master configuration file for this configuration is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsmconfig SYSTEM "fmsmasterconfig.dtd">
<fmeworld>
    <fmenterprise 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" />
        </fmenterprise>
    </fmeworld>
```

- **FSC configuration files**

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="fscl"/>
</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>
```

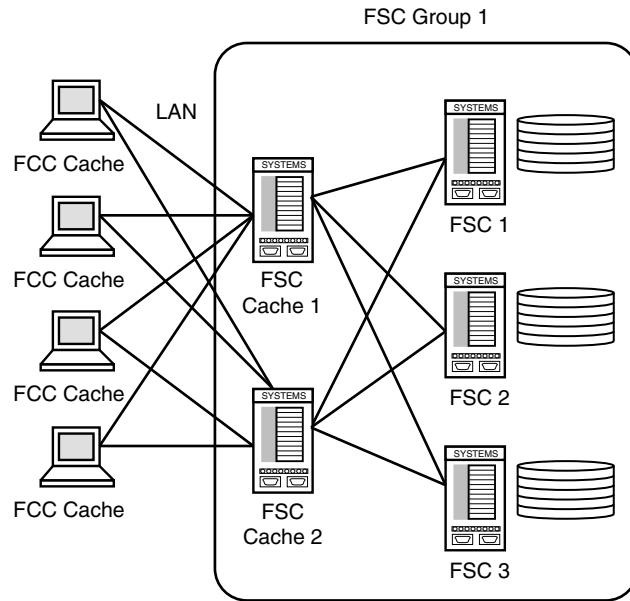
- **FCC configuration file**

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="rlnx03.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>
```

FCC LAN client failover configuration

This configuration provides a hot local LAN failover configuration.



- **Master configuration file**

This file is configured similar to the previous examples. 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 fail over machines to cover the unlikely event of a three-box failure scenario.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsmconfig SYSTEM "fmsserverconfig.dtd">
<fmsserver>
    <fmselement id="fms.teamcenter.com">
        <fccdefaults>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fccdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$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**

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 fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>
</fsccconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="true" />
    <fsc id="fsc2"/>
</fsccconfig>
FSC3
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>

```

- **FCC configuration file**

This example of the FCC configuration file is similar to previous examples. The key point is:

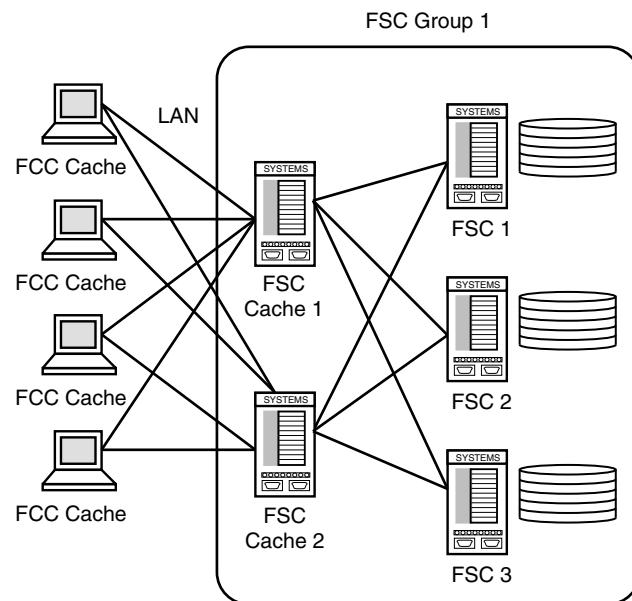
Configuration failover

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 **parentfscs**.



- **Master configuration file**

This file is configured similar to the previous examples. The key points of this configuration's configuration file are:

dnszone clientmap attribute

The **dnszone** attribute can be used instead of the **subnet** and **mask** attributes to map FSCs.

dnshostname clientmap attribute

The **dnshostname** attribute can be used instead of the **subnet** and **mask** attributes to map a specific host to FSCs.

default clientmap 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 clientmap 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 "fscmasterconfig.dtd">
<fmeworld>
    <fmenterprise id="fms.teamcenter.com">
        <fscdefaults>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fscdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$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" />
            <assignedfsc fscid="fscCache2" priority="1" />
        </clientmap>
        <clientmap dns_not_defined="true">
            <assignedfsc fscid="fscCache1" priority="0" />
            <assignedfsc fscid="fscCache2" priority="1" />
        </clientmap>
    </fscGroup>
</fmenterprise>
</fmeworld>
```

- **FSC configuration files**

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"/>
```

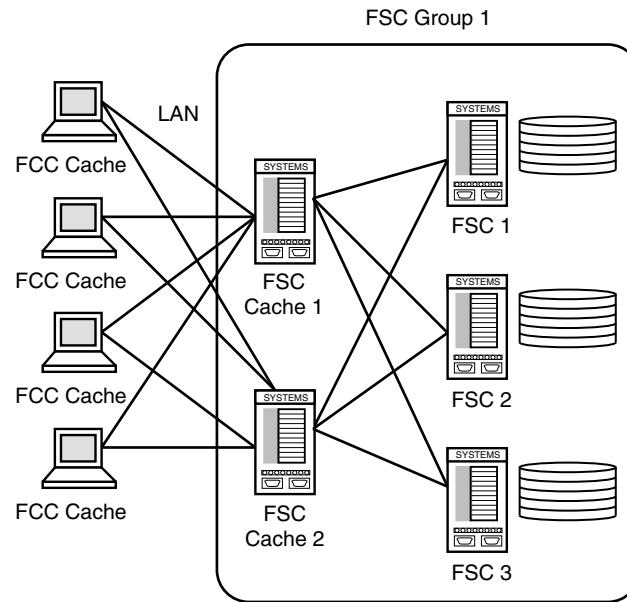
```

</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 fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>

```

FCC external load balancing configuration

This configuration illustrates how to load balance FMS data.



- **Master configuration file**

This file is configured similar to the previous examples. The key points of this configuration's configuration file are:

assignedfsc priority attribute

Within the master 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 fsccconfig SYSTEM "fsccconfig.dtd">
<fsccworld>
    <fmserprise 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>
</fmserprise>
</fmsworld>

```

- **FCC configuration file**

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**.

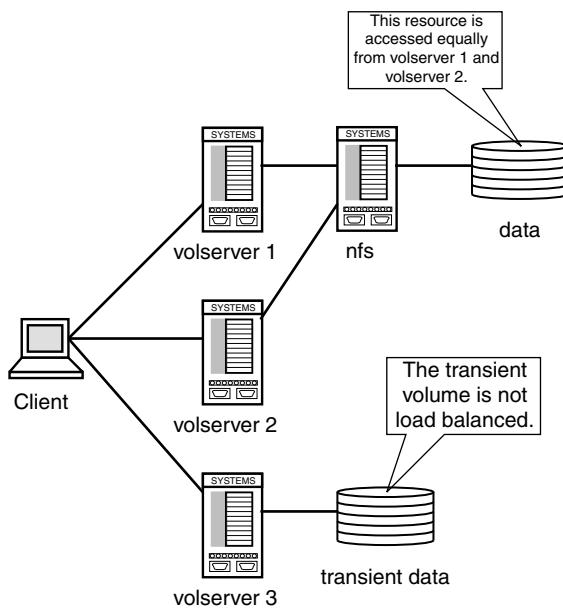
```

<?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.



- **Master configuration file**

This file is configured similar to the previous examples. 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 "fmssmasterconfig.dtd">
<fmeworld>
    <fmenterprise 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>
        </fscGroup>
    </fmenterprise>
</fmeworld>
```

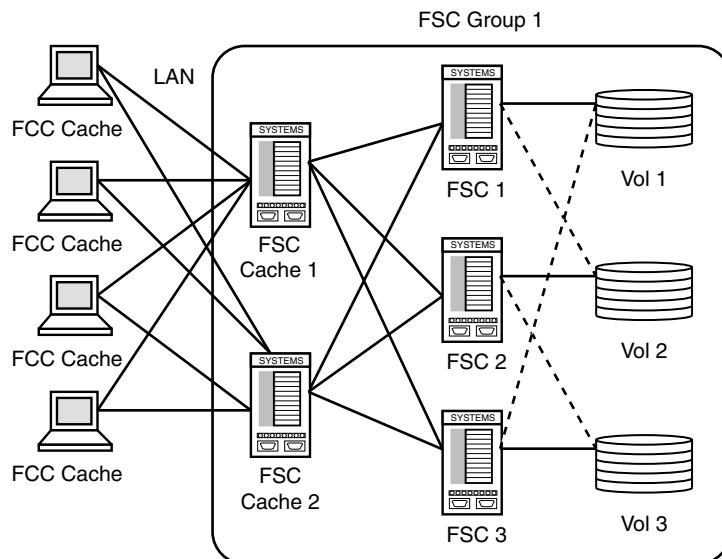
```

        </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>
</fmseenterprise>
</fmeworld>

```

FSC volume failover configuration

This configuration illustrates how to configure an FSC volume failover.



- **Master configuration file**

This file is configured similar to the previous examples. 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 fsmconfig SYSTEM "fmssmasterconfig.dtd">
<fmeworld>
    <fmseenterprise id="fms.teamcenter.com">
        <fscdefaults>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fscdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$USER/FCCCache"
                overridable="true"/>
        </fccdefaults>
    </fmseenterprise>
</fmeworld>

```

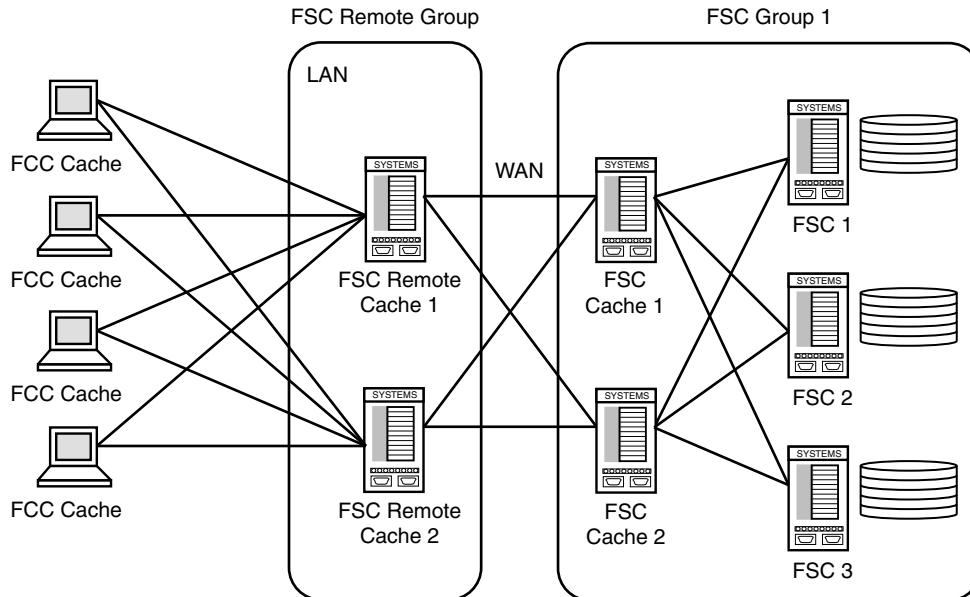
```

<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>
</fmseenterprise>
</fmeworld>

```

FSC remote cache failover configuration

This configuration provides a hot remote cache configuration, and a idle central cache configuration.



- **Master configuration file**

This file is configured similar to the previous examples. 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 fsmconfig SYSTEM "fmsmasterconfig.dtd">
<fmeworld>
    <fmenterprise id="fms.teamcenter.com">
        <fscdefaults>
            <property name="FSC_ReadCacheLocation"
                      value="$HOME/FscCache1/tmp/FscCache"
                      overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                      value="$HOME/FscCache1/tmp/FscCache"
                      overridable="true"/>
        </fscdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                      value="$HOME/FCCCache1/tmp/$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>
            <entryfsc fscid="fscache1" priority="0" />
            <entryfsc fscid="fscache2" priority="1" />
        </fscGroup>
        <linkparameters fromgroup="fscRemoteGroup"
                        togroup="fscGroup1"
                        transport="wan">
        </linkparameters>
    </fmenterprise>
</fmeworld>
```

- **FSC configuration files**

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**

This example of the FCC configuration file is similar to previous examples.

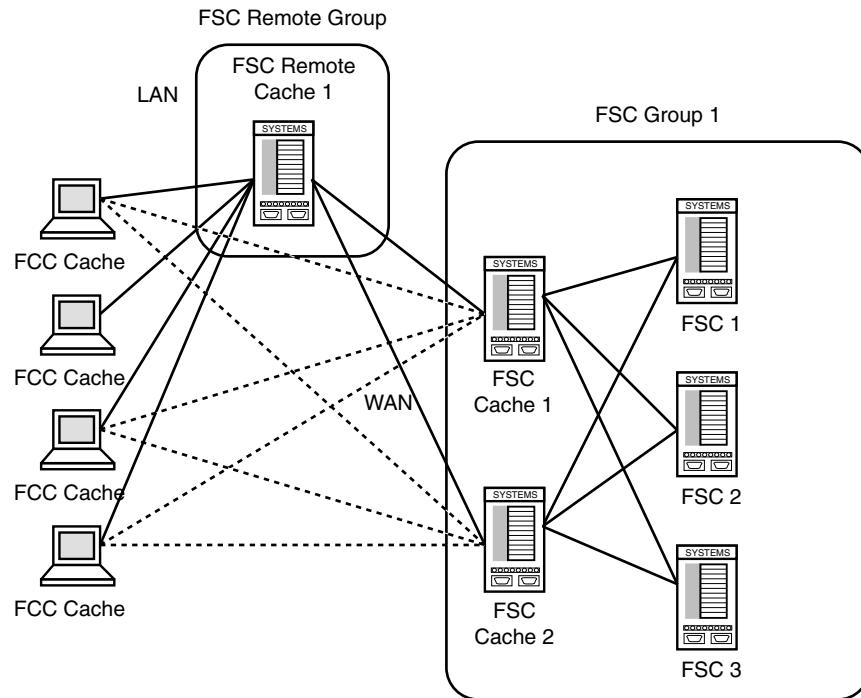
```

<?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.



- Master configuration file**

This file is configured similar to the previous examples. 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

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

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 "fmssmasterconfig.dtd">
<fmeworld>
    <fmenterprise id="fms.teamcenter.com">
        <fscdefaults>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fscdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$USER/FCCCache"
                overridable="true"/>
        </fccdefaults>
        <fsc 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>
        </fsc>
        <fscGroup id="fscGroup1">
            <fsc id="fscCache1" address="http://csun15.ugs.com:4444" />
            <fsc id="fscCache2" address="http://csun16.ugs.com:4444" />
            <fsc id="fscl" address="http://csun17.ugs.com:4444">
                <volume id="vol1" root="/data/vol1"/>
            </fsc>
        </fscGroup>
    </fmenterprise>
</fmeworld>
```

```
<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>
</fmeworld>
```

- **FSC configuration files**

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>
```

```
</fscconfig>
```

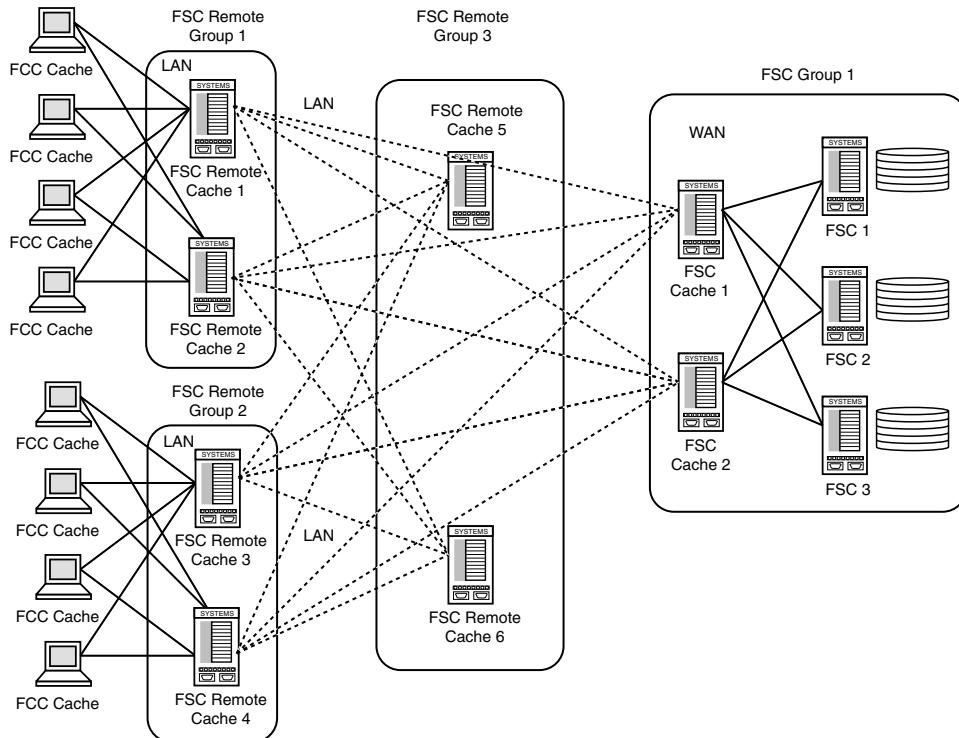
- **FCC configuration file**

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 fail over for either a single point of failure, or fail over if both of the FSC Remote Group 3 cache machines fail.



- **Master configuration file**

This file is configured similar to the previous examples. 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 fsmconfig SYSTEM "fmsmasterconfig.dtd">

<fmeworld>
    <fmenterprise id="fms.teamcenter.com">
        <fscdefaults>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fscdefaults>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$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">
            </linkparameters>
        </linkparameters>
    </fmsenterprise>
</fmeworld>
```

- **FSC configuration files**

This FSC configuration file for this configuration is as follows:

```

FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>
</fsccconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc2"/>
</fsccconfig>
FSC3
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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**

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"/>

```

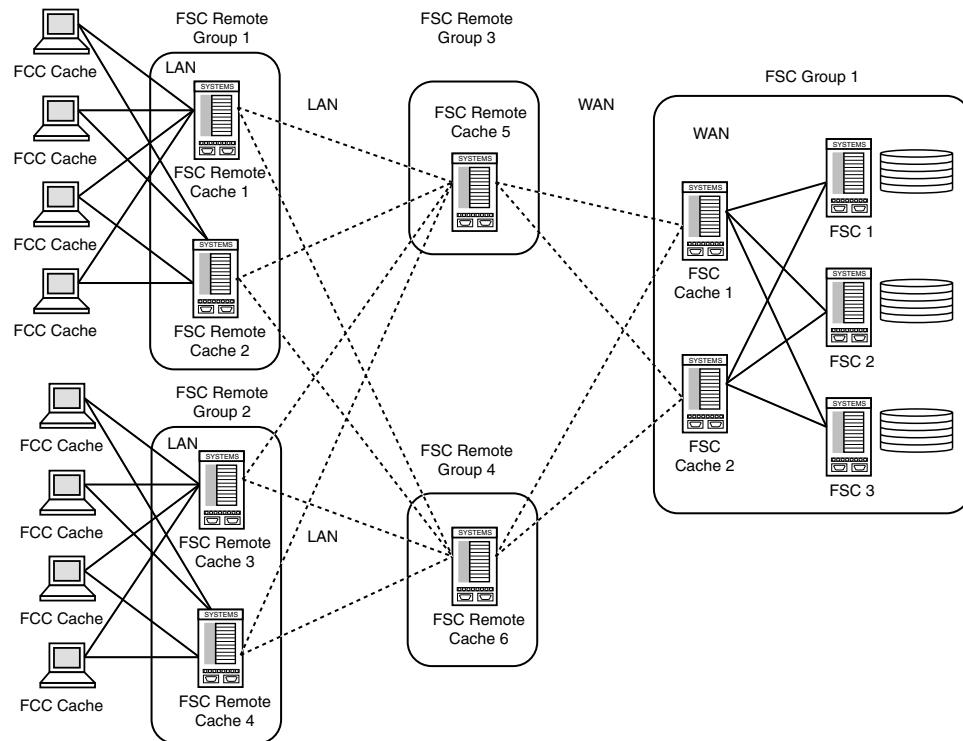
```

<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.



- Master configuration file**

This file is configured similar to the previous examples. 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 "fmssmasterconfig.dtd">

<fmeworld>
    <fmenterprise id="fms.teamcenter.com">
        <fscdecls>
            <property name="FSC_ReadCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
            <property name="FSC_WriteCacheLocation"
                value="$HOME/FscCache1/tmp/FscCache"
                overridable="true"/>
        </fscdecls>
        <fccdefaults>
            <property name="FCC_CacheLocation"
                value="$HOME/FCCCache1/tmp/$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>
    </fmenterprise>
</fmeworld>
```

```

                        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">
            </linkparameters>
        </fmsenterprise>
    </fmeworld>

```

- FSC configuration files**

This FSC configuration file for this configuration is as follows:

```

FSC1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="true"/>
    <fsc id="fsc1"/>
</fsccconfig>
FSC2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <fscmaster serves="false" />
    <fsc id="fsc2"/>
</fsccconfig>
FSC3
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>
FscCache2
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fsccconfig SYSTEM "fsccconfig.dtd">
<fsccconfig>
    <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"/>
</fsccconfig>

```

```

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**

This example of the FCC configuration file is similar to previous examples.

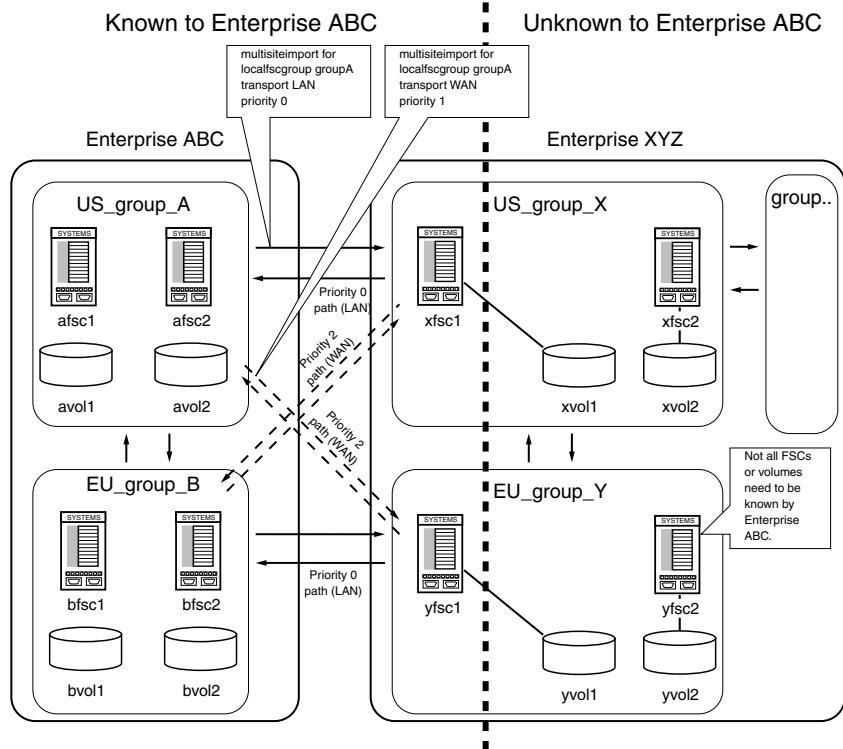
```

<?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.



- **Master configuration file**

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. 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

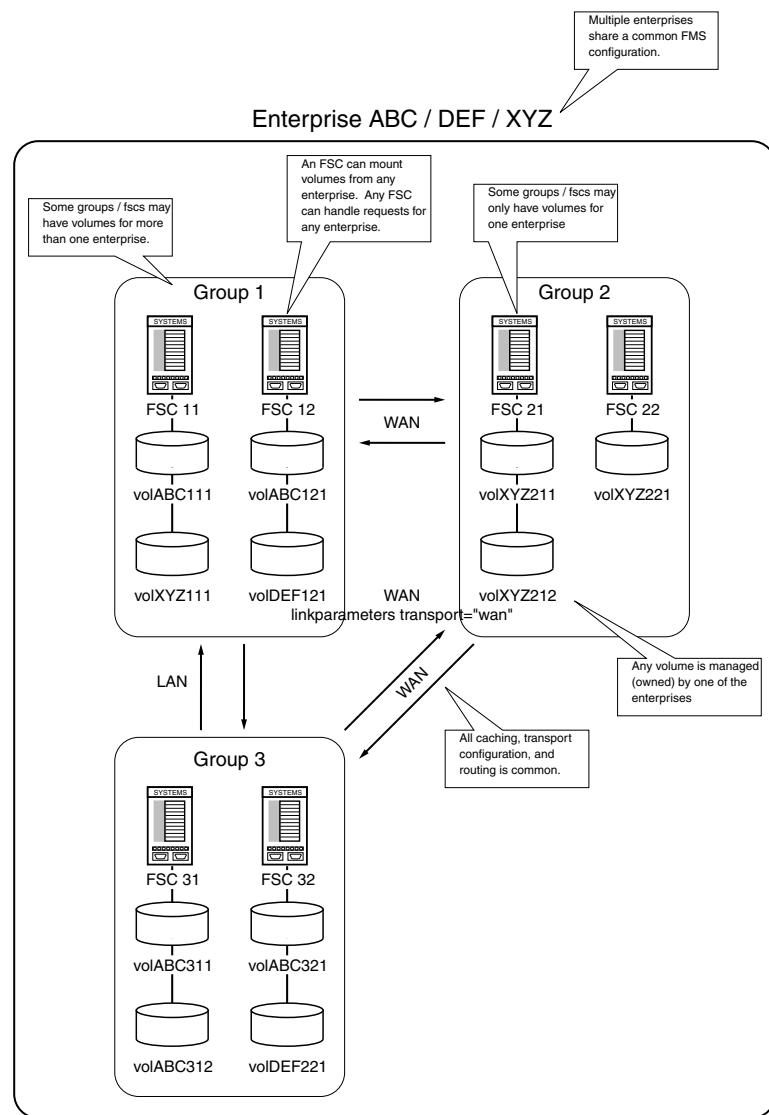
The **wan** attribute allows you to direct remote traffic via the WAN transport mode. Use this attribute to configure WAN routes between geographically distant sites.

```
<fmeworld>
<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>
<fmenterprise 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>
</fmenterprise>
</fmeworld>
```

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.



- **Master configuration file**

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. 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.

```
<fmeworld>
  <fmenterprise id="ABC">
```

```
<fmenterprise id="DEF"/> <!-- DEF is enterprise site -->
<fmenterprise 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"/>
    <accesson 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"/>
</fmenterprise>
</fmeworld>
```

Chapter

9 *Configuring Teamcenter for performance*

Configuring the four-tier architecture for performance	9-1
Introduction to configuring the four-tier architecture for performance	9-1
Managing server memory	9-2
Introduction to managing server memory	9-2
Default automatic memory cleanup settings	9-2
Determining optimum settings for a two-tier environment	9-3
Determining optimum settings for a four-tier environment	9-4
Disabling automatic memory cleanup	9-4
Server memory cleanup log files	9-4
Shared memory	9-7
Shared memory for metadata	9-7
Introduction to shared memory for metadata	9-7
Configuring the shared memory cache	9-8
Synchronizing the shared memory cache	9-8
Shared memory for Text Server data	9-9
Introduction to shared memory for Text Server data	9-9
Remove memory backing store files	9-9
Define a fixed directory for shared memory map files	9-10
Shared memory for preferences	9-10
Shared memory for localized LOV data	9-11
Define a fixed directory for shared memory map files	9-11
Sharing a TcServer instance with multiple applications	9-12
Tuning the Web tier	9-12
Tuning application servers	9-12
Start the administrative interface	9-13
Web tier monitoring	9-14
Configure monitoring with the webtierMonitorConfig.xml file	9-15
Sample webtierMonitorConfig.xml code	9-17
Configuring the rich client for startup performance	9-18
Introduction to configuring the rich client for startup performance	9-18
Configuring the FCC file warmer	9-20
Configure the filewarmer.properties file	9-20
Configure the filelist.txt file	9-20
Configure file warmer logging behavior	9-21
Configure the FCC to locate the filewarmer.properties file	9-21
Configuring TCCS to start when users log on to a Windows' operating system	9-22
Setting PATH and AUX_PATH for enhanced performance	9-25

Cleaning the POM_timestamp table	9-26
Cleaning the backpointer table after upgrade	9-26

Chapter

9 *Configuring Teamcenter for performance*

Configuring the four-tier architecture for performance

Introduction to configuring the four-tier architecture for 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.

Additional 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.

For more information about setting this preference, see the [Frequently Asked Questions for Teamcenter](#).

For more reference information about this preference, see the [Preferences and Environment Variables Reference](#).

- 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 **TC_ROOT\pool_manager\serverPool.properties** file.

For more information, see [Pool-specific configuration tuning](#).

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.

For information about changing the default configuration of memory cleanup of unused objects, see [Determining optimum settings for a two-tier environment](#) and [Determining optimum settings for a four-tier environment](#).

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.
UNLOAD_TRIGGER_FLOOR	5000	Specifies (in kilobytes) what memory size must be reached to initiate the unloading of unused objects from the server memory.

Preference	Default value	Description
UNLOAD_MINIMUM_LIFETIME	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. For more information about memory cleanup logging, see <i>Server memory cleanup log files</i> .

For information about changing the default configuration of memory cleanup of unused objects, see *Determining optimum settings for a two-tier environment* and *Determining optimum settings for a four-tier environment*.

Determining optimum 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 setting are adequate for most two-tier environments, including the default Teamcenter installation running only the **Foundation** template.

Determining optimum 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 logon.
- 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 of 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 operating from initiating.
Siemens PLM Software recommends using this method to disable automatic memory cleanup.
- Setting the **UNLOAD_MINIMUM_LIFETIME** 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** 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.

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.
-------------------	--

Level 2

End-session report	<p>Specifies the peak memory used by unloadable objects for the entire session.</p> <p>Specifies the least amount of memory used by unloadable objects for the entire session.</p> <p>Specifies the total number of objects unloaded.</p> <p>Specifies the total number of unloading operation requests.</p> <p>Specifies the total number of actual unloading operations.</p> <p>Specifies the average time of an unloading operation.</p>
--------------------	---

Level 3

In-session report	<p>Specifies the number of objects in the cache.</p> <p>Specifies the number of objects removed from cache.</p> <p>Specifies the time (in seconds) to unload the cache.</p> <p>Specifies total memory usage of all areas (in bytes) before unloading and before callbacks. Usage shown by area.</p> <p>Specifies total memory usage of all areas (in bytes) after unloading and before callbacks. Usage shown by area.</p> <p>Specifies the number of objects remaining in the cache.</p>
-------------------	---

Setting the **UGII_CHECKING_LEVEL** preference 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	<p>Specifies the peak memory used by unloadable objects for the entire session.</p> <p>Specifies the least amount of memory used by unloadable objects for the entire session.</p> <p>Specifies the total number of objects unloaded.</p> <p>Specifies the total number of unloading operation requests.</p> <p>Specifies the total number of actual unloading operations.</p> <p>Specifies the average time of an unloading operation.</p> <p>Provides a summary of unloading per type, including type name and unload count.</p>
--------------------	--

Level 4

In-session report	<p>Specifies, if adding to the cache: Object String =object string , type: type of object, time: when it was accessed</p> <p>Specifies, if removing from the cache: Unloading Object: Object String =object string , type: type of object, time: when it was accessed</p> <p>Specifies the number of objects in the cache.</p> <p>Specifies the number of objects removed from cache.</p> <p>Specifies the time (in seconds) to unload the cache.</p> <p>Specifies total memory usage of all areas (in bytes) before unloading and before callbacks. Usage shown by area.</p> <p>Specifies total memory usage of all areas (in bytes) after unloading and before callbacks. Usage shown by area.</p> <p>Specifies the number of objects remaining in the cache.</p>
End-session report	<p>Setting the UGII_CHECKING_LEVEL preference to 1 provides the following callback and memory usage information:</p> <p>Callbacks. (Lists callbacks triggered. Specifies the time (in seconds) callbacks took.)</p> <p>Specifies memory usage of all areas (in bytes) before callbacks after unloading. Usage shown by area.</p> <p>Specifies memory usage of all areas (in bytes) after callbacks after unloading. Usage shown by area.</p> <p>Specifies the number of objects remaining in the cache.</p> <p>Specifies the time (in seconds) to complete the call.</p> <p>Specifies the peak memory used by unloadable objects for the entire session.</p> <p>Specifies the least amount of memory used by unloadable objects for the entire session.</p> <p>Specifies the total number of objects unloaded.</p> <p>Specifies the total number of unloading operation requests.</p> <p>Specifies the total number of actual unloading operations.</p> <p>Specifies the average time of an unloading operation.</p> <p>Provides a summary of unloading per type, including type name and unload count.</p>

Shared memory

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.

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.
For more information, see the [Business Modeler IDE Guide](#).
- From TEM, select the **Generate Server Cache** check box in the **Foundation Settings** panel.
For more information, see the [Teamcenter Environment Manager Help](#).

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 UNIX. By default, the service/daemon starts automatically.

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).
For more information, see the [Preferences and Environment Variables Reference](#).
- 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.

For more information, see the [Preferences and Environment Variables Reference](#).

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.

For more information, see the [Utilities Reference](#).

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 UNIX 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)*. For more information, see [Remove memory backing store files](#).

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.

The UNIX operating system does not relinquish semaphores when processes are complete. If new locations are routinely created for shared memory mapping files, the semaphore count eventually reaches the UNIX operating system limit. The following message displays and the system reverts to in-process memory functionality.

```
ACE_Malloc_T<ACE_MEM_POOL_2, ACE_LOCK, ACE_CB>::ACE_Malloc_T: Not enough space
```

If you are running on a UNIX platform, you can ensure you do not exceed the semaphore limit by defining a fixed directory for shared memory mapping files. For more information, see [Define a fixed directory for shared memory map files](#).

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** (UNIX) 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**.

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 UNIX 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.

Shared memory for preferences

Shared memory for preferences connects all Teamcenter servers for a particular version to the shared memory created by the first Teamcenter server. When using a four-tier configuration, changes to preferences for four-tier configuration immediately affect all existing Teamcenter server processes on a host.

This functionality creates a backing store file with a **.mem** extension. The backing store file contains data for preferences definitions, default values, and overlay values. If the file is deleted, another is automatically created by the next Teamcenter server process. The backing store file is stored in the **TC_TMP_DIR/TC_VERSION_STRING/db_siteID/Preferences** directory.

This functionality is implemented with the **TC_USE_PREFS_SHARED_MEMORY** environment variable, which is turned **ON** by default. For more information about this environment variable, see the *Preferences and Environment Variables Reference*.

Note When changes are made from a different host using the same database, the changes are updated on that host, but not on the first host. The next new Teamcenter server processes begun on the first host updates the shared memory segment and the changes affect all the processes on that host.

For example, consider a pool of servers in which two are running. One of these two servers creates a preference with a protection scope of **Site**. In this situation, the other running server does not see the preference until the rich client is refreshed. Any servers that start after the preference was created see the new preference immediately.

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 **Fnd0LOVDisplayAsEnabled** global constant to disable loading localized LOV display names.

For more information, see the *Business Modeler IDE Guide*.

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 UNIX 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.

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 is 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, WebLogic, and so on) 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 or EAR 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. For example, the **weblogic.xml** file is unique to WebLogic, and is where application JSP and servlet check intervals are specified.

Each major version of the JRE (for example, Java 5.0) has unique GC tuning options available.

For more information about tuning JRE, see the documentation at:

http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html

You can use third-party applications to view Web tier administration interface data in a more comprehensive manner.

For more information, see *Using third-party applications to view server manager administration data*.

Start the administrative interface

The Web application collects a variety of metrics on application events that are relevant to performance and sizing.

1. Extract a copy of the **pref_export.xml** file from the **mldcfg.jar** file in the deployed **.ear** file and place it in the root directory of the application server. (For example, place it on the domain directory in WebLogic or the profile directory in WebSphere.)

Alternatively, you can find a copy of the **pref_export.xml** file in the **\$TC_ROOT/pool_manager** directory.

2. Start the Web application server.
3. In the **pref_export.xml** file, set **RunHtmlAdapter** to **true**.

4. Open a Web browser and type **http://application server host:8092**.

Note You can change the port number in the **pref_export.xml** file if necessary.

5. Log on using the default user name and password **plmmonitor** and **localhost**, respectively.

The Web application metrics appear.

6. Click any of the links for the listed metric MBean.

Web tier monitoring

Web tier monitoring provides notification when the following metrics reach the specified level, indicating a critical event:

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.

For more information about using the XML file, see [Configure monitoring with the webtierMonitorConfig.xml file](#).

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 e-mail notifications are generated from the **/webtierMonitorConfigInfo.xml** file. (This is a companion file to the **webtierMonitorConfig.xml** file.) For a complete list of possible causes and recommended actions for Web tier monitoring, see this file.

This file is stored in the EAR file by default. A copy can be placed on the application sever's current directory to override the default version.

Configure Web tier metrics and logging behavior to better administer Web tier processes.

Enable Web tier metrics in the **pref_export.xml** file. This file is stored in the EAR file by default. A copy can be placed on the application sever's current directory to override the default version.

- Use the **RunHtmlAdapter** element to run the JMX HTML adapter for the Web tier upon startup.
- 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 EAR file, application sever startup directory, or someplace on the application server **classpath** (allowing settings to be tailored to a machine if required).
2. 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 e-mail 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 e-mail responder. Multiple e-mail responders can be configured, in which case the identifiers must be unique.
 - **protocol**
Specify the e-mail protocol by which notifications are sent. SMTP is the only supported protocol.
 - **hostAddress**

Specify the server host from which the e-mail 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 e-mails are sent.

- **toAddress**

Specify the address to which the notification e-mails are sent. You can specify multiple e-mail addresses, separated by semi-colons.

- **suppressionPeriod**

Specify the amount of time (in seconds) to suppress e-mail notification of critical events.

For more information, see the suppression period example in [Introduction to monitoring](#).

- **emailFormat**

Specify the format in which the e-mail 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.

For more information, see the suppression period example in [Introduction to monitoring](#).

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's monitoring mode to one of the following:

- **Collect**

Collect metric data and display results in the MBean view for this metric.

This is the default setting.

- **Alert**

Collect metric data, display results in the MBean view (within the server manager administrative interface) for this metric, and send e-mail notifications when critical events exceed the specified threshold.

- **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.

Sample webtierMonitorConfig.xml code

In the following example, two of the four monitoring metrics are configured for e-mail notification of critical events. And two **EmailResponder** elements are configured. E-mail notification is sent to **EmailResponder1** when the **No Server Available** metric achieves critical event status. E-mail notification is sent to **EmailResponder2** when the **Grave Events** metric achieves critical event status. Data is collected for the other metrics, but no e-mail 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>
    </Metric>
  </MetricsConfig>
</ApplicationConfig>
```

```

        <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

Introduction to 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 operation 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 **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 nonlocal folders from the **PATH** statement.

For more information, see [Setting PATH and AUX_PATH for enhanced performance](#).

- 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 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 **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 UNIX 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 PLM 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.

For more information, see [*Configuring the FCC file warmer*](#).

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

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 PLM Software recommends setting the following paths:

```
@include  
RAC-install-path\portal\plugins  
RAC-install-path\portal\features  
RAC-install-path\portal\registry  
RAC-install-path\portal\configuration  
RAC-install-path\portal\Teamcenter.exe  
RAC-install-path\portal\Teamcenter.ini  
RAC-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.

- UNIX 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** (UNIX) 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.

- UNIX systems:

```
-Dfilewarmer.properties=/usr/bin/Teamcenter/fcc/filewarmer.properties
```

Use the full path to the properties file. Use single forward slashes as directory separators.

Configuring 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

On Windows Vista and later (including Windows 7), 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 PLM 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 .

For more information about running the **fccstat -kill** command, see method 2 in *Shutting down a TCCS/FCC instance*. For more information about working with Windows shutdown scripts, see the Microsoft documentation at:

<http://technet.microsoft.com/en-us/library/cc753404.aspx>

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:\Program~1\Siemens\Teamcenterversion-number\tccs
```

C:\Program~1\Siemens\Teamcenterversion-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 1.6 or later.

For example:

```
set JRE_HOME=C:\Program~2\Java\jre6
```

- 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_FCCSTARTUPLOG=C:\fccstartup.log
```

- d. Start TCCS.

```
call %FMS_HOME%\bin\fccstat -start
```

FMS_HOME is the value set in step 1.

- 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 PLM 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:\Program~1\Siemens\Teamcenter9\tccs  
set JRE_HOME=C:\Program~2\Java\jre6  
set FMS_FCCSTARTUPLOG=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/Vista/Server2008

- Single-user system:

**C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Startup**

- Multi-user system:

**C:\Users\user-name\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup**

Windows XP/2000

- Single-user system:

**C:\Documents and Settings\All Users\Start
Menu\Programs\Startup**

- Multi-user system:

**C:\Documents and Settings\user-name\Start
Menu\Programs\Startup**

Note These directories may be hidden by default.

- (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.

- (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 PATH and AUX_PATH 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%
```

- UNIX 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**.

Cleaning the POM_timestamp table

Each time an object is modified during a Teamcenter session, a timestamp record is created. Typically, records older than the time specified by the **TC_TIMESTAMP_THRESHOLD** environment variable are deleted from the table when a user logs off. However, if an operation continues after a user logs off, or there is an error during the session, records remain in the table.

The table must be periodically cleaned of these accumulated records. Specify how often the table is cleaned using the **TC_TIMESTAMP_THRESHOLD** environment variable.

By default, this environment variable is set to **96** hours (four days). The optimum cleaning time varies by site. Variables include how quickly table size grows at your site and user requirements. For example, if there are users at your site who must be logged on consecutively for many days, the setting must be increased.

Note As of Teamcenter 9.1, timestamps of modified objects are stored in the **POM_TIMESTAMP** table as well as the **PPOM_OBJECT** table. (Previously, this timestamp information was stored only in the larger **PPOM_OBJECT** table.) Storing timestamp records in the **POM_TIMESTAMP** table enhances product performance.

Cleaning the backpointer table after upgrade

As of Teamcenter 9.1, **relation_type** object references and **ImanRelation** primary and secondary object references are no longer stored in the backpointer table. They are stored only in the **ImanRelation** table.

Note Where-referenced queries now search the **ImanRelation** table for **ImanRelation** references, rather than searching the backpointer table.

This significant reduction in the size of the backpointer table can improve product performance. To take advantage of this performance improvement, you must run the **clean_backpointer** utility on your Teamcenter database after upgrading from a previous version to Teamcenter 9.1 (or a later version). This utility is not run during upgrade, as the cleanup operation may be time-consuming.

The 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 varies from site to site, depending on infrastructure elements such as database load, network performance, server configuration, and so on. Because performance varies, Siemens PLM Software recommends following these best practices:

1. Run the utility with the **-m** argument set to **INFO** to determine the number of objects stored in the backpointer table.
2. Run the utility with the **-s** argument set to a few thousand objects and note how long it takes to delete the objects.
3. Use the results of these first two operations to determine the length of time it takes to clear the entire backpointer table (**-s=ALL**) and schedule accordingly.

For more information about the utility, see the [*Utilities Reference*](#).

Chapter

10 Logging

Introduction to logging	10-1
Using the Log Manager	10-1
Logging for business logic servers	10-3
System log files	10-3
Configuring business logic server logging	10-3
Configure logging with the logger.properties file	10-4
Debugging using business logic server logging	10-5
Logging for Teamcenter tiers	10-5
Overview of logging for Teamcenter tiers	10-5
Client tier logging	10-8
Web tier logging	10-8
Enterprise tier logging	10-18
Resource tier logging	10-24
Translation server	10-27

Chapter

10 Logging

Introduction to logging

Log files are generated in each Teamcenter tier, as well as in third-party applications used to provide Teamcenter capabilities. Logging is comprised of the following components:

- Log Manager
 - Provides a mechanism to consolidate log files generated across the Teamcenter deployment.
 - For more information, see [Using the Log Manager](#).
- System log files
 - Provide system-level logging from the business logic layer.
 - For more information, see [System log files](#).
- Teamcenter tier log files
 - Provide logging generated in each Teamcenter tier.
 - For more information, see [Overview of logging for Teamcenter tiers](#).

Using the Log Manager

You can examine error log files to troubleshoot problems. Log files are generated in each Teamcenter tier, as well as in third-party applications used to provide Teamcenter capabilities. The Log Manager provides a mechanism for you to consolidate log files that are generated across the Teamcenter deployment. Set the log volume location where the Log Manager writes logs using the **TC_ROOT/fsc/log.properties** file.

The Log Manager:

- Captures log files on a local disk. (NFS mounts can be used for log files that are not performance intensive.)
- Accepts all log files, including legacy logs in any format, and logs based on a standard set of loggers.
- Queries for software that uses the standard set of loggers.
- Writes all log files to a location that is managed by the Log Manager software.
- Performs log management functions such as query, scanning, and purging.

- Uses TPTP transport over RMI for delivery of the log files.
- Supports both process logs and task-based logs (where a transaction is executed on behalf of a specific user request).

Log files are divided into two types: *task logs* and *process logs*. Task logs represent the output of long-running service processes, such as those found in the Dispatcher Server system. Task logs are stored in a directory named by the GUID or task ID and can be distributed on many computers. Without Log Manager this requires you to search all the different log directories based on a GUID. The Log Manager provides the capability to search all task logs deployed throughout the system for a specific task's log files, or for all failed log files. This supports analysis of translation failure to identify and correct root causes.

Any log file that is not based on a task ID is considered a process log. This includes most process log files such as **syslog** files, FSC logs, audit logs, and so on, which may include log records on behalf of multiple users performing any number of tasks. The Log Manager accepts and provide access to all process logs that are placed in a log volume. Service processes are configured to write log information to a specific log volume. The Log Manager supports type designations for the various process log files to enable you to search for and retrieve specific types of log files.

The Log Manager employs a log writer to capture log files to a local or mounted log volume directory. The log writer software provides interfaces for capturing task and process logs as well as metadata for each log file. Log file metadata includes information such as the log file completion status, the host and process name that captured the log file, and the log file type. The log writer writes log data and log metadata directly to either local or mounted disk (log volume). The primary function of the Log Manager service is to query and retrieve log file metadata for display in an administrator's or user's interface. The Log Manager provides general query interface for metadata such as completion status or query by a specific task ID.

The benefits of the Log Manager are:

- Direct to disk log capture

Efficient capture to disk is essential to avoid a negative impact on the performance of critical system functions. Once captured, logs may be searched or loaded into databases as appropriate.

- Centralized access

Although logs are captured in a globally distributed manner, users and administrators can view the accumulated logs together to understand overall system operation.

- Common interfaces

A standard set of log capture APIs, file formats, and log retrieval APIs is provided to simplify the process of log monitoring.

- Integration with third-party vendors

Producing logs on a single infrastructure in a specific set of logging volumes, and with a single retrieval API, enables third-party vendors to quickly integrate logging and monitoring software.

Logging for business logic servers

System log files

All system log files provide the following information:

- Priority level
- Date/time (UTC format)
- Log correlation ID of the client request
- Error code (when applicable)
- Message
- Logger name
- Caller file and line number (if specified)

You can dynamically change logging levels for the system log file.

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.
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.

You must configure loggers to write messages of the desired priority level. Setting a logger at **DEFAULT** causes it to inherit its priority level from its parent logger.

For more information about configuring logging levels, see [Configuring business logic server logging](#).

Configuring business logic server logging

There are two methods available to manage logging levels for business logic servers.

- You can make persistent changes to logging levels of business logic servers using the **logger.properties** file, which is stored in the *TC_DATA* directory. Changing logging levels in this file affects all servers in the server pool. If multiple pools use the *TC_DATA* directory, all servers in all server pools using

this directory are affected. This method is useful for updating deployment environments.

For information about using this file, see [Configure logging with the logger.properties file](#).

Note Changes to the file take effect only after the server is restarted. You can use the **Restart Warm Servers** button in the server manager administrative interface to restart all warm servers and implement the changes to the logging levels.

For more information about using the **Restart Warm Servers** button in the J2EE server manager administrative interface, see [Administering the pool's server manager](#).

For more information about using the **Restart Warm Servers** button in the .NET server manager administrative interface, see [Restarting warm servers](#).

- You can dynamically change logging levels for a particular user session using the J2EE server manager administrative interface.
Changing logging methods in this manner affects only the selected business logic server, and the changes last only throughout the user session.
For information about using this method, see [Configure business logic server manager logging in the J2EE server manager administrative interface](#).

Configure logging with the logger.properties file

The **logger.properties** file lists all loggers used by the business logic servers.

1. Open the *TC_DATA/logger.properties* file.
2. Change the logging level of any logger:
 - a. Scroll to the logger whose logging level you want to change.
 - b. Type a valid logging level after the equal (=) sign.
Setting a logger at **DEFAULT** causes it to inherit its priority level from its parent logger.
For more information about logging levels, see [Server manager logging levels](#).
- c. Choose **File→Save** to save your changes.

Note Changes to the file take effect only after the server is restarted. You can use the **Restart Warm Servers** button in the server manager administrative interface to restart all warm servers and implement the changes to the logging levels.

For more information about using the **Restart Warm Servers** button in the J2EE server manager administrative interface, see [Administering the pool's server manager](#).

For more information about using the **Restart Warm Servers** button in the .NET server manager administrative interface, see [Restarting warm servers](#).

Debugging using business logic server logging

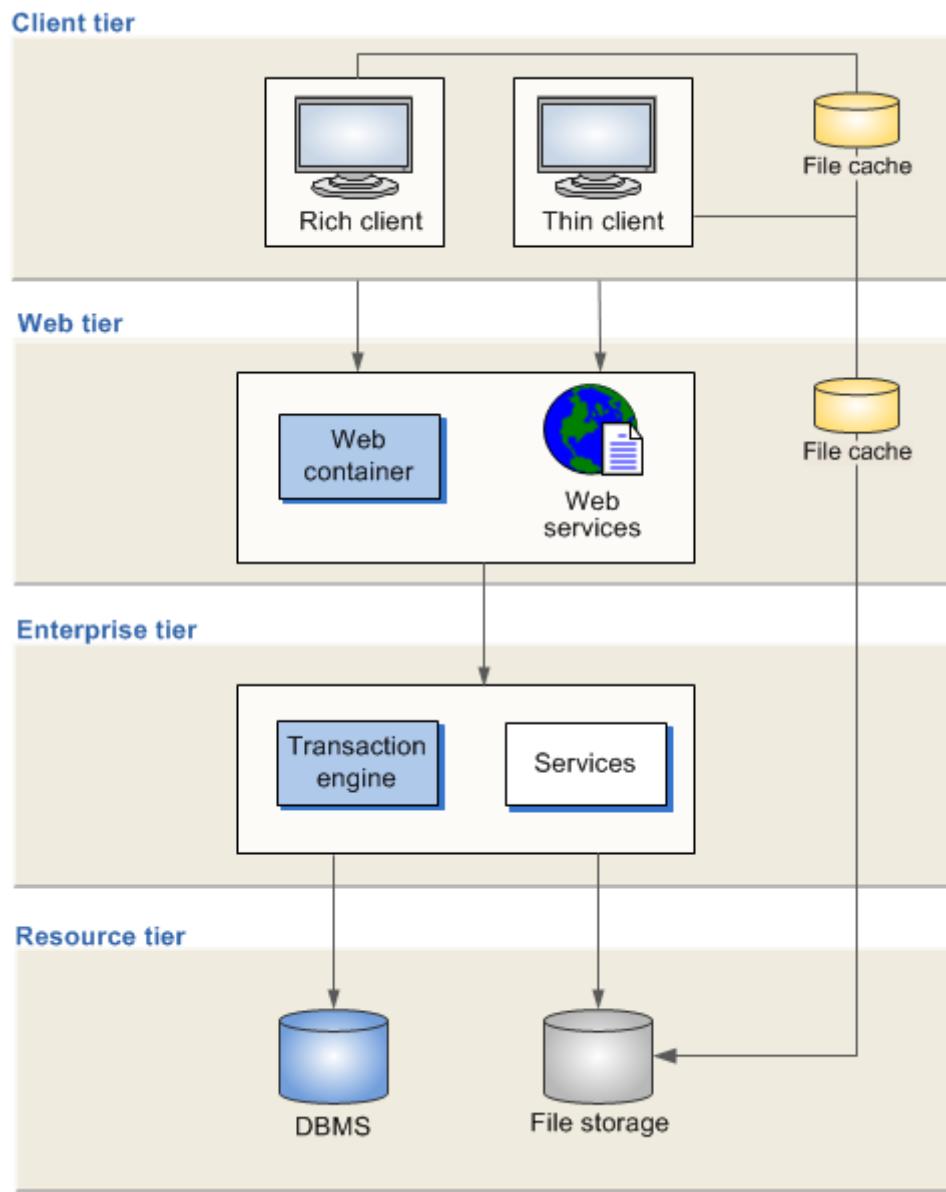
There are two methods available for setting business logic server logging for debugging.

- Set the **UGII_CHECKING_LEVEL** preference to **1** to enable server checking. When checking is enabled, the system uses the **logger.debug.properties** file for logging instead of the **logger.properties** file. The default settings of the debug file generate useful debugging messages.
- Caution**
- Enabling checking significantly increases the size of log files. Only enable checking for debugging purposes or when requested by Siemens PLM Software support.
- Set the **TC_LOGGER_CONFIGURATION** environment variable to the whole file path of a properties file or the path of the directory containing the **logger.properties** file to use for debugging. You can specify a custom logger properties file or the **TC_DATA/logger.debug.properties** file.

Logging for Teamcenter tiers

Overview of logging for Teamcenter tiers

Log files are generated in each Teamcenter tier. To understand the purpose of log files produced by different tiers in the Teamcenter architecture, a review of the architecture is necessary.



The four-tier architecture comprises the following tiers:

- Client tier
The client tier hosts client applications, provides user interface input and output processing, and hosts secure file caches.
- Web tier
The Web tier routes client requests to business logic, serves static content to clients, and processes login requests.
- Enterprise tier
The enterprise tier hosts business logic, applies security rules, and serves dynamic content to clients.
- Resource tier

The resource tier stores persistent data (bulk and metadata).

The log correlation ID is a unique ID that records the path of a service request starting from the Web tier through the enterprise tier. This log correlation ID is recorded in the log messages on each tier that processes the request. The log correlation ID for the browser-based client has the following structure:

- Client-or-Proxy-Host.Unique-ID.User-Name.Request-Count*
- *Client-or-Proxy-Host* indicates the client host name or the proxy server host name.
 - *Unique-ID* is a unique, randomly generated value.
 - *User-Name* is the user name associated with the request. The default is set to **Anonymous**.
 - *Request-Count* is a counter that gets incremented for each request.

In the browser-based client, the client sends the request from the Web browser and this request is received first by the Web tier. The **WebTier.log** file records the request with the correlation ID as follows:

```
DEBUG - cmh6p199.net.plm.eds.com.05340.Anonymous.00002 -
2011/04/21-02:00:38,223 UTC - ci6p199 - Begin - WebClientPreProcess
[com.teamcenter.presentation.webclient.actions.WebClientPreProcess.
handleAction(WebClientPreProcess.java:226):Thread[[ACTIVE]
ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)
',5,Pooled Threads]]
...
...
```

The same log correlation ID is recorded in **ServerManager.log** file, where the server manager assigns the **tcserver1** process for this user.

```
DEBUG - cmh6p199.net.plm.eds.com.05340.Anonymous.00002 -
2011/04/21-02:02:20,921
UTC - cmh6p199 - Server assigned: tcserver1@poolA@5920@ci6p199
[com.teamcenter.jeti.serversubpoolmanager.ServerPoolManager$Assign
er.publishAssignment(ServerPoolManager.java:7934):Thread[RequestPr
ocessor-40,10,main]]
```

The request path can be traced in the **tcserver1.syslog** file using the same log correlation ID.

```
2011-04-20 22:03:03 cmh6p199.net.plm.eds.com.05340.Anonymous.00002 -
Service Request: T2LWebMethodService:process request raw
Successfully loaded dynamic module D:\udu\ref\tc9.0.0.2011041800\wnti32\lib\libweb.dll
Loaded module D:\udu\ref\tc9.0.0.2011041800\wnti32\lib\libvis.dll 129c0000 90000
3cc8fcd-4dd74f64-12a895a6-4c835394-1=libvis_1303160187 version = 9000.0.0.4104
Loaded module D:\udu\ref\tc9.0.0.2011041800\wnti32\lib\libweb.dll 125000000 4a8000
65af7f91-4e213a95-24a44db3-b6e6544-1=libweb_1303161766 version = 9000.0.0.4104
INFO - 2011/4/21-02:03:03.190 UTC - cmh6p199.net.plm.eds.com.05340.Anonymous.00002 -
Loaded library libweb - Teamcenter.Soa.TcServerUtil
tcscript took 0.078000s cpu, 0.085000s real to parse file - toplevel.html
```

On the next client request, this log correlation ID is updated with the authenticated user name and an incremented request counter.

```
DEBUG - cmh6p199.net.plm.eds.com.05340.infodba.00003 -
2011/04/21-02:03:04,777 UTC - cmh6p199 - Begin - WebClientPreProcess
[com.teamcenter.presentation.webclient.actions.WebClientPreProcess.
handleAction(WebClientPreProcess.java:226):Thread[[ACTIVE]
ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)
',5,Pooled Threads]]
```

Subsequent client requests use the same log correlation ID with an incremented request counter until the client logs off or the client Web session times out.

Client tier logging

The rich client is a Java client hosted in the Eclipse framework. It is installed using a URL. An automatic bootstrap ensures the latest approved configuration is running.

The thin client is an AJAX-style client with DTML/JavaScript-based rendering. It is supported by Internet Explorer and Firefox. No separate client installation is required.

Rich client logging	
Component	Rich client
Description	Captures the client events. When Debug is turned ON , a correlation ID is written to the log file.
Log file	\$user-name_TcRAC.log
Location	<i>java.io.tmpdir</i> This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	The log4j.appenders.TcLoggerFileAppender.file variable in the TC_INSTALL/portal/plugins/configuration_release/TcLogger.properties file.

There is no thin client logging.

Web tier logging

The Web tier is the client's gateway into the server system. It exposes the services provided by the enterprise tier, providing the routing to the correct server for each request and performing authentication and authorization checks.

JBoss logging	
Component	JBossWeb server
Description	Logs transaction messages to log files stored within the application server directory structure.
Log file	WebTier.log
Location	jboss-4.0.5GA\bin\logs\WebTier\process\
Configuration	Specify the LogVolumeLocation file by regenerating the tc.ear file. Run F:\tc_web\insweb.bat , select Modify , and then select Modify Context Parameters . Set the LogVolumeLocation and click OK . The new tc.ear is generated. Deploy it in the application server. This creates a WebTier directory under LogVolume .

Component	JBossWeb server
-----------	------------------------

Description	Contains console messages from the application implementing MLD.
Log file	plmconsole.txt
Location	jboss-4.0.5GA\bin\logs\MLD\process\
Configuration	N/A

Component	JBossWeb server
Description	Contains sampled gauge values and threshold notifications. This local file is overridden by any JMX interface configuration.
Log file	RTEvents.txt
Location	jboss-4.0.5GA\bin\logs\MLD\process\
Configuration	N/A

Component	JBossWeb server
Description	Default log file for response time monitoring.
Log file	RTConsole.txt
Location	jboss-4.0.5GA\bin\logs\MLD\process\
Configuration	N/A

Component	JBossWeb server
Description	Contains notifications processed by remote Response Time GaugeListeners and TraceListeners .
Log file	RTRemotes.txt
Location	jboss-4.0.5GA\bin\logs\MLD\process\
Configuration	N/A

Component	JBossWeb server
Description	Local file for response time tracing messages and logging.
Log file	RTTraces.txt
Location	jboss-4.0.5GA\bin\logs\MLD\process\
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the WebTier.log
Log file	WebTier.log.xml

Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the plmconsole.txt
Log file	plmconsole.txt.xml
Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the RTEvents.txt
Log file	RTEvents.txt.xml
Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the RTConsole.txt
Log file	RTConsole.txt.xml
Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the RTRemotes.txt .
Log file	RTRemotes.txt.xml
Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

Component	JBossWeb server
Description	Metadata file for the RTTraces.txt
Log file	RTTraces.txt.xml
Location	jboss-4.0.5GA\bin\logs\WebTier\metadata\process
Configuration	N/A

WebLogic logging	
Component	WebLogic Web server

WebLogic logging	
Description	Logs transaction messages to log files stored within the application server directory structure.
Log file	WebTier.log
Location	bea\user_projects\domains\teamcenter\logs\WebTier\process\
Configuration	Specify the LogVolumeLocation file by regenerating the tc.ear file. Run F:\tc_web\insweb.bat , select Modify , and then select Modify Context Parameters . Set the LogVolumeLocation and click OK . The new tc.ear is generated. Deploy it in the application server. This creates a WebTier directory under LogVolume .

Component	WebLogic Web server
Description	Contains console messages from the application implementing MLD.
Log file	plmconsole.txt
Location	bea\user_projects\domains\teamcenter\logs\MLD\process\
Configuration	N/A

Component	WebLogic Web server
Description	Contains sampled gauge values and threshold notifications. This local file is overridden by any JMX interface configuration.
Log file	RTEvents.txt
Location	bea\user_projects\domains\teamcenter\logs\MLD\process\
Configuration	N/A

Component	WebLogic Web server
Description	Default log file for response time monitoring
Log file	RTConsole.txt
Location	bea\user_projects\domains\teamcenter\logs\MLD\process\
Configuration	N/A

Component	WebLogic Web server
Description	Contains notifications processed by remote Response Time GaugeListeners and TraceListeners .

Log file	RTRemotes.txt
Location	bea\user_projects\domains\teamcenter\logs\MLD\process\
Configuration	N/A

Component	WebLogic Web server
Description	Local file for response time tracing messages and logging
Log file	RTTraces.txt
Location	bea\user_projects\domains\teamcenter\logs\MLD\process\
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the WebTier.log
Log file	WebTier.log.xml
Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the plmconsole.txt
Log file	plmconsole.txt.xml
Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the RTEvents.txt
Log file	RTEvents.txt.xml
Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the RTCConsole.txt
Log file	RTCConsole.txt.xml

Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the RTRemotes.txt
Log file	RTRemotes.txt.xml
Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

Component	WebLogic Web server
Description	Metadata file for the RTTraces.txt
Log file	RTTraces.txt.xml
Location	bea\user_projects\domains\teamcenter\logs\WebTier\metadata\process
Configuration	N/A

WebSphere logging	
Component	WebSphere Web server
Description	Logs transaction messages to log files stored within the application server directory structure.
Log file	WebTier.log
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\
Configuration	Specify the LogVolumeLocation file by regenerating the tc.ear file. Run F:\tc_web\insweb.bat , select Modify , and then select Modify Context Parameters . Set the LogVolumeLocation and click OK . The new tc.ear is generated. Deploy it in the application server. This creates a WebTier directory under LogVolume .

Component	WebSphere Web server
Description	Contains console messages from the application implementing MLD.
Log file	plmconsole.txt
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\MLD\process
Configuration	N/A

Component	WebSphere Web server
Description	Contains sampled gauge values and threshold notifications. This local file is overridden by any JMX interface configuration.
Log file	RTEvents.txt
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\MLD\process
Configuration	N/A

Component	WebSphere Web server
Description	Default log file for response time monitoring
Log file	RTConsole.txt
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\MLD\process
Configuration	N/A

Component	WebSphere Web server
Description	Contains notifications processed by remote Response Time GaugeListeners and TraceListeners .
Log file	RTRemotes.txt
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\MLD\process
Configuration	N/A

Component	WebSphere Web server
Description	Local file for response time tracing messages and logging
Log file	RTTraces.txt
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\MLD\process
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the WebTier.log
Log file	WebTier.log.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the plmconsole.txt
Log file	plmconsole.txt.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the RTEvents.txt
Log file	RTEvents.txt.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the RTCConsole.txt
Log file	RTCConsole.txt.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the RTRemotes.txt
Log file	RTRemotes.txt.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Component	WebSphere Web server
Description	Metadata file for the RTTraces.txt
Log file	RTTraces.txt.xml
Location	IBM\WebSphere\AppServer\profiles\AppSrv01\logs\WebTier\process\metadata\
Configuration	N/A

Oracle logging	
Component	Oracle application server

Oracle logging	
Description	Logs transaction messages to log files stored within the application server directory structure.
Log file	WebTier.log
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	Specify the LogVolumeLocation file by regenerating the tc.ear file. Run F:\tc_web\insweb.bat , select Modify , and then select Modify Context Parameters . Set the LogVolumeLocation and click OK . The new tc.ear is generated. Deploy it in the application server. This creates a WebTier directory under LogVolume .

Component	Oracle application server
Description	Contains console messages from the application implementing MLD.
Log file	plmconsole.txt
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	N/A

Component	Oracle application server
Description	Contains sampled gauge values and threshold notifications. This local file is overridden by any JMX interface configuration.
Log file	RTEvents.txt
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	N/A

Component	Oracle application server
Description	Default log file for response time monitoring
Log file	RTConsole.txt
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	N/A

Component	Oracle application server
Description	Contains notifications processed by remote Response Time GaugeListeners and TraceListeners .
Log file	RTRemotes.txt
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	N/A

Component	Oracle application server
Description	Local file for response time tracing messages and logging
Log file	RTTraces.txt
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\process\
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the WebTier.log
Log file	WebTier.log.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\WebTier\metadata\process
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the plmconsole.txt
Log file	plmconsole.txt.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\MLD\metadata\process
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the RTEvents.txt
Log file	RTEvents.txt.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\MLD\metadata\process
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the RTCConsole.txt
Log file	RTCConsole.txt.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\MLD\metadata\process
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the RTRemotes.txt

Log file	RTRemotes.txt.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\MLD\metadata\process
Configuration	N/A

Component	Oracle application server
Description	Metadata file for the RTTraces.txt
Log file	RTTraces.txt.xml
Location	oracle\10.1.3\OracleAS\j2ee\home\logs\MLD\metadata\process
Configuration	N/A

.NET logging without Log Manager	
Component	.NET Web server
Description	Contains Web tier logs. The logs are written to the Windows event logs.
Log file	TcWebTier.evtx
Location	C:\Windows\System32\winevt\Logs
Configuration	N/A

Enterprise tier logging

The enterprise tier hosts the business logic, making queries and performing transactions for the clients, managing access control on product data, and serving dynamic content to the clients.

Server manager logging	
Component	Server manager
Description	Contains messages from the server manager application.
Log file	ServerManager.log
Location	TC_ROOT/pool_manager/logs/ServerManager/process
Configuration	Increase the information logged to this file by increasing the severity level of the TC_ROOT/pool_manager/log4j.xml file. Change the location of this file by resetting the LogVolumeLocation value of the TC_ROOT/pool_manager/log.properties file.

Component	Teamcenter server
Description	Metadata file for the ServerManager.log

Log file	ServerManager.log.xml
Location	TC_ROOT/pool_manager/logs/ServerManager/metadata/process
Configuration	N/A

Teamcenter server logging	
Component	Teamcenter server
Description	Diagnoses errors. Captures information, errors and warnings.
Log file	tcserverpid.syslog
Location	TC_TMP_DIR This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA/tc_profilevars.bat file. For information about configuring logging levels, see <i>Configuring business logic server logging</i> .

Component	Teamcenter server
Description	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.
Log file	tcserverpid.jnl
Location	TC_TMP_DIR This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA/tc_profilevars.bat file.

Component	Teamcenter server
Description	Tracks actions performed on objects at a session level, such as folder creation.
Log file	tcserverpid.log
Location	TC_TMP_DIR This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA/tc_profilevars.bat file.

Component	Teamcenter server
-----------	-------------------

Description	Captures information regarding CORBA ORB server information and transactions issued by the TAO ORB in the server.
Log file	tcserverpid.orblog
Location	<p>TC_TMP_DIR</p> <p>This value is typically C:\Temp on Windows and /tmp on UNIX.</p>
Configuration	<p>Define the TC_TMP_DIR environment variable in the TC_DATA/tc_profilevars.bat file.</p> <p>With the default value, 0, set for the ORB Log Level setting, information is not written to the log and the log file is automatically removed at the end of a successful session.</p> <p>Change the logging level for a particular server by changing the TAO Log Level value using the J2EE server manager administrative interface.</p> <p>For more information, see <i>Configure business logic server manager logging in the J2EE server manager administrative interface</i>.</p> <p>Change the logging level for all new servers by adding a -ORBDebugLevel level clause to the SERVER_PARAMETERS pool specific property.</p> <p>For more information, see <i>Pool-specific configuration tuning recommendations</i>.</p>

Component	Teamcenter server
Description	Contains information regarding the attempted access to unauthorized data. The information includes failed logon events and attempts to access unauthorized objects in the database.
Log file	security.log
Location	<p>TC_LOG</p> <p>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.</p>
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA/tc_profilevars.bat file.

Component	Teamcenter server
Description	Tracks Teamcenter installation messages. The date-time stamp represents the date and time Teamcenter Environment Manager was run. For example, install0522241627.log indicates that Teamcenter Environment Manager was run at 4:27 on February 24, 2005.
Log file	install.log

Location	TC_TMP_DIR
This value is typically C:\Temp on Windows and /tmp on UNIX.	
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA\tc_profilevars.bat file.

Component	Teamcenter server
Description	Contains the standard output from the POM utilities called by Teamcenter Environment Manager.
Log file	pomutilities.log
Location	TC_TMP_DIR
	This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	Define the TC_TMP_DIR environment variable in the TC_DATA\tc_profilevars.bat file.

Component	Teamcenter server
Description	Tracks changes made to system objects such as users, groups, volumes, and so on. Also tracks system events such as releasing objects.
Log file	administration.log
Location	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.
Configuration	N/A

Component	Teamcenter server
Description	Contains entries regarding platform operation, such as Teamcenter startup and shutdown events.
Log file	system.log
Location	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.
Configuration	N/A

Component	Teamcenter server
-----------	-------------------

Description	Tracks selected properties for specified actions in the database. These audit logs are created in Audit Manager.
Log file	audit.log
Location	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.
Configuration	N/A

Component	Teamcenter server
Description	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.
Log file	\$FSC_ID_startup.log on UNIX. %FSC_ID%stdout.log and %FSC_ID%stderr.log on Windows.
Location	/tmp on UNIX. %FMS_HOME% on Windows.
Configuration	N/A

PLM XML logging	
Component	PLM XML
Description	Provides complete information regarding the current PLM XML export or import.
Log file	xml-file-name.log or plmxml_log_timestamp.log
Location	TC_TMP_DIR This value is typically C:\Temp on Windows systems and /tmp on UNIX systems. 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 rich client export, the log file is generated at the same location as the XML file.
Configuration	Determine the logging level with the PLMXML_log_file_content preference.

Multi-Site logging	
Component	Multi-Site

Multi-Site logging	
Description	Tracks actions performed on objects at a session level, such as imported or exported objects.
Log file	idsmID.log
Location	TC_TMP_DIR on the machine hosting the IDSM server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Component	Multi-Site
Description	Diagnoses errors. Captures information, errors and warnings.
Log file	idsmID.syslog
Location	TC_TMP_DIR on the machine hosting the IDSM server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Component	Multi-Site
Description	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.
Log file	idsmID.jnl
Location	TC_TMP_DIR on the machine hosting the IDSM server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Component	Multi-Site
Description	Tracks actions performed on objects at a session level.
Log file	odsID.log
Location	TC_TMP_DIR on the machine hosting the ODS server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Component	Multi-Site
Description	Diagnoses errors. Captures information, errors and warnings.
Log file	odsID.syslog

Location	TC_TMP_DIR on the machine hosting the ODS server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Component	Multi-Site
Description	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.
Log file	odsID.jnl
Location	TC_TMP_DIR on the machine hosting the ODS server. This value is typically C:\Temp on Windows and /tmp on UNIX.
Configuration	N/A

Resource tier logging

The resource tier stores persistent data, such as the database where product data is stored and managed files. It also stored administrative data, including user data in LDAP-compliant repositories.

File Management System (FMS) logging	
Component	FMS
Description	Contains log entries regarding server run-time operations.
Log file	\$FSC_ID_startup.log on UNIX. %FSC_ID%stdout.log and %FSC_ID%stderr.log on Windows.
Location	/tmp on UNIX. %FMS_HOME% on Windows.
Configuration	Determine the logging level by setting \$FSC_HOME/FSC_\$FSC_ID_\$USER.xml in the fsc.xml file on UNIX. (FMS resolves \$HOME to %USERPROFILE% on Windows.) Valid values are FATAL , ERROR , WARN , INFO , and DEBUG . Siemens PLM Software recommends that you never run an FSC in debug mode. Generally, WARN and INFO provide sufficient logging information.

Component	FMS
Description	Contains log entries regarding server run-time operations.
Log file	\$FSC_ID.log
Location	FSC_HOME/logs/FSC/process

Configuration	Change the location of this file by setting the LogVolumeLocation property in the <i>FSC_HOME/log.properties</i> file.
---------------	---

Component	FMS
Description	Default console log file for response time logging
Log file	FSC_host-name_user-ID_plmconsole.txt
Location	FSC_HOME/logs/MLD/process
Configuration	N/A

Component	FMS
Description	Contains sampled gauge values and threshold notifications. This file is overridden by any JMX interface configuration.
Log file	FSC_host-name_user-ID_RTEvents.txt
Location	FSC_HOME/logs/MLD/process
Configuration	N/A

Component	FMS
Description	Contains response time tracing messages and logging.
Log file	FSC_host-name_user-ID_RTTraces.txt
Location	FSC_HOME/logs/MLD/process
Configuration	N/A

Component	FMS
Description	Contains notifications processed by remote Response Time GaugeListeners and TraceListeners .
Log file	FSC_host-name_user-ID_RTRemotes.txt
Location	FSC_HOME/logs/MLD/process
Configuration	N/A

Component	FMS
Description	Metadata file for the FSC_host-name_user-ID_fsc.log
Log file	FSC_host-name_user-ID_fsc.log.xml
Location	FSC_HOME/logs/MLD/metadata/process
Configuration	N/A

Component	FMS
-----------	-----

Description	Metadata file for the FSC_host-name_user-ID_RTEvents.txt
Log file	FSC_host-name_user-ID_RTEvents.txt.xml
Location	FSC_HOME/logs/MLD/metadata/process
Configuration	N/A

Component	FMS
Description	Metadata file for the FSC_host-name_user-ID_RTTraces.txt
Log file	FSC_host-name_user-ID_RTTraces.txt.xml
Location	FSC_HOME/logs/MLD/metadata/process
Configuration	N/A

Component	FMS
Description	Metadata file for the FSC_host-name_user-ID_RTRemotes.txt
Log file	FSC_host-name_user-ID_RTRemotes.txt.xml
Location	FSC_HOME/logs/MLD/metadata/process
Configuration	N/A

Component	FMS
Description	Metadata file for the FSC_host-name_user-ID_RTConsole.txt
Log file	FSC_host-name_user-ID_RTConsole.txt.xml
Location	FSC_HOME/logs/MLD/metadata/process
Configuration	N/A

Business Modeler IDE logging	
Component	Business Modeler IDE
Description	Contains deployment messages.
Log file	deploy.log
Location	output/deploy/serverProfileName/date-timestamp
Configuration	N/A

Component	Business Modeler IDE
Description	Contains tcfs logging messages.
Log file	Migration logs
Location	output/migration
Configuration	N/A

Translation server

The translation server asynchronously distributes translation requests to machines with the resource capacity to execute the requests. A grid technology manages the job distribution, communication, translator execution, security, and error handling for translation requests. Translation requests are triggered based by workflow, data checkin, batch mode, or on-demand operations.

Translation server logging	
Component	Translation server
Description	Contains Translation Module messages while processing the task.
Log file	<i>task-ID_m.log</i>
Location	<i>LogVolumeDirectory/TSTK/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Contains Translation Scheduler messages while processing the task.
Log file	<i>task-ID_s.log</i>
Location	<i>LogVolumeDirectory/TSTK/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Contains Translation Service messages while processing the task. The Translation Service receives translation task requests from the client and sends them to the translation server.
Log file	<i>task-ID_ts.log</i>
Location	<i>LogVolumeDirectory/TSTK/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Contains Scheduler messages.
Log file	Scheduler.log
Location	<i>LogVolumeDirectory/TSTK/process</i>
Configuration	N/A

Component	Translation server
Description	Contains Module messages.
Log file	Module_ID.log

Location	<i>LogVolumeDirectory/TSTK/process</i>
Configuration	N/A

Component	Translation server
Description	Contains Adminclient messages.
Log file	Adminclient.log
Location	<i>LogVolumeDirectory/TSTK/process</i>
Configuration	N/A

Component	Translation server
Description	Contains a history of events and of state transitions performed on all the tasks.
Log file	History.log
Location	<i>LogVolumeDirectory/TSTK/process</i>
Configuration	N/A

Component	Translation server
Description	Contains Translation Service process messages such as startup and cleanups .
Log file	TranslationService.log
Location	<i>LogVolumeDirectory/TSTK/process</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the <i>task-ID_m.log</i> file
Log file	<i>task-ID_m.log.xml</i>
Location	<i>LogVolumeDirectory/TSTK/metadata/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the <i>task-ID_s.log</i> file
Log file	<i>task-ID_s.log.xml</i>
Location	<i>LogVolumeDirectory/TSTK/metadata/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the task-ID_ts.log file
Log file	task-ID_ts.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/task/task-ID</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the Scheduler.log file
Log file	Scheduler.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/process</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the Module_ID.log file
Log file	Module_ID.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/process</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the AdminClient.log file
Log file	AdminClient.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/process</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the History.log file
Log file	History.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/process</i>
Configuration	N/A

Component	Translation server
Description	Metadata file for the TranslationService.log file
Log file	TranslationService.log.xml
Location	<i>LogVolumeDirectory/TSTK/metadata/process</i>
Configuration	N/A

Chapter

11 Backing up and recovering files

Overview of the backup and recovery process	11-1
Oracle Recovery Manager (RMAN)	11-3
Introduction to the Oracle Recovery Manager (RMAN)	11-3
Benefits of RMAN	11-4
Features of RMAN	11-5
ARCHIVELOG mode considerations	11-6
Restoring purged files	11-7
Single file recovery (SFR)	11-7
Single file recovery object model	11-7
Single file recovery in Teamcenter rich client	11-8
Single file recovery query	11-9
Using alternative hot backup and recovery procedures	11-9
Back up and restore database files	11-9
Back up and restore Teamcenter volumes	11-10
Use Virtual Device Interface (VDI)	11-11

Chapter

11 *Backing up and recovering files*

Overview of the backup and recovery process

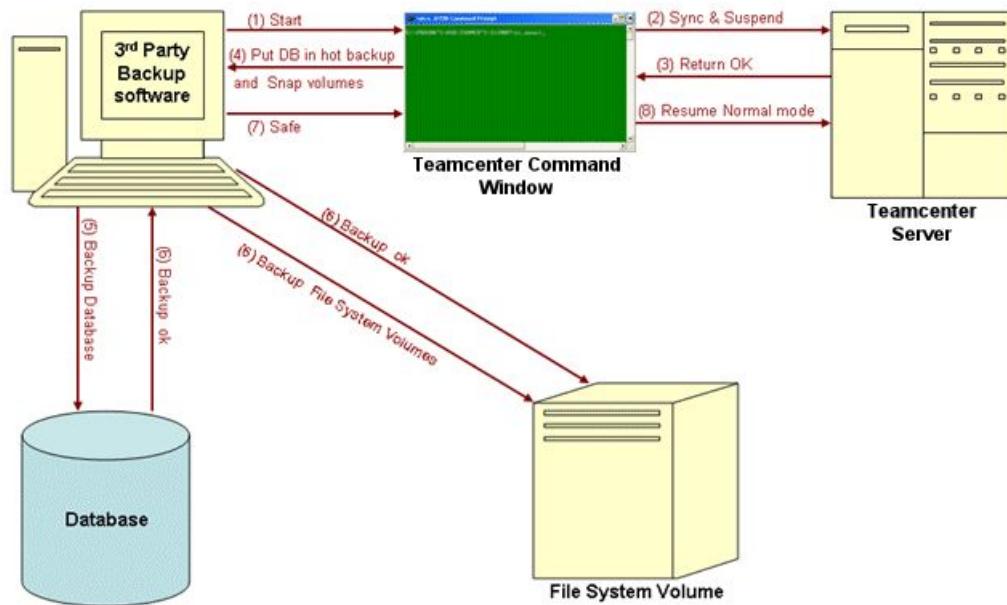
The integrated backup and recovery feature facilitates third-party backup systems to perform online backup, allowing Teamcenter to operate continually. This functionality focuses on the area of backing up metadata and math data, and recovering that data in different restoration scenarios. To accomplish this, the integrated backup feature places Teamcenter in different operation modes using the **backup_modes** utility.

The integrated backup system operates in three modes: **Read-Only**, **Blobby Volume**, and **Normal**. These are the different modes prompted in the rich client.

Mode	Description
Read-Only Mode	Places Teamcenter into read-only state. This state holds writing files to the volume during backup.
Blobby Volume Mode	Places Teamcenter in blobby (temporary) volume mode. Teamcenter can be switched into this mode after the third-party backup software takes a snapshot of the data, thus allowing continuous availability.

Normal Mode

Places Teamcenter back in normal mode from read-only, or blobby volume mode.



The following steps describe the process flow of a third-party integrated backup:

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 e-mail 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.
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 bobby 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 PLM 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 PLM 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, MS SQL, DB2).
- All database volumes.
- The *TC_DATA* directory.

- The **TC_ROOT\install** directory, which stores configuration data.
- The **TC_ROOT\bmide** directory, which 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 backups of a NOARCHIVELOG database after a consistent shutdown.	Backs up all blocks, not just the changed blocks. Does not allow you to recover a NOARCHIVELOG database.
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.

Recovery Manager	User-managed method
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: <ul style="list-style-type: none">• Database schema at a specified time.• 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	Does not include any reporting functionality.

Features of RMAN

Feature	Description
Incremental backups	Up to four levels; level 0 and levels 1 and 4 .

Feature	Description
Corrupt block detection	<p>During backup:</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Limits number of open files. • Size of backup piece.
Recovery catalog	Automates restore and recovery operations.
Selective backups	Backs up an entire database, selected tablespaces, or selected datafiles.

Note RMAN was introduced in Oracle version 8.0 and is not compatible with Oracle databases prior to version 8.0. For more information about Oracle's RMAN, see the following URL:

http://download.oracle.com/docs/cd/B19306_01/backup.102/b14193.pdf

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 math data 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 PLM Software recommends using SFR to recover a single file.

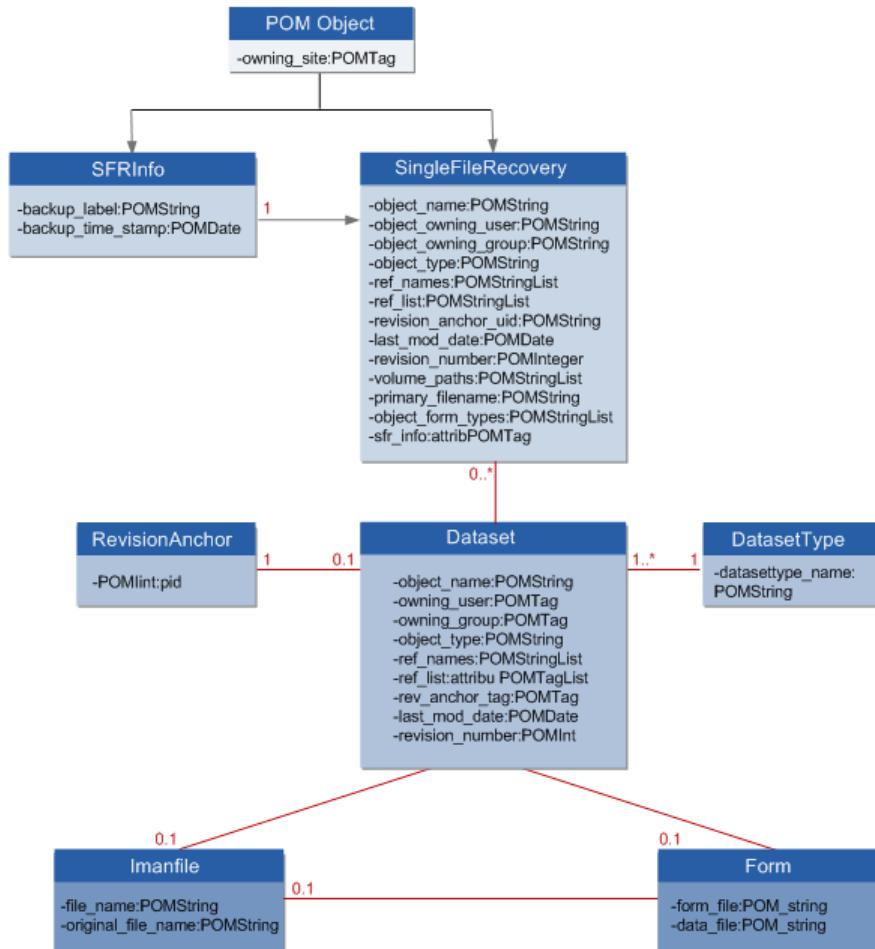
SFR uses File Management System (FMS) to search for and restore files within the time limits specified by the **TC_sfr_recovery_interval** and **TC_sfr_process_life_time** preferences. The third-party backup software recovers the purged versions of files from the Teamcenter volume. If the file is found, 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_Label,    /**<(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. For additional information, see the *Preferences and Environment Variables Reference*.

- **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. For more information, see the *Preferences and Environment Variables Reference*.

Using alternative hot backup and recovery procedures

Back up and restore database files

If you do not use third-party storage management systems, Siemens PLM 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 2000/2005 supports online (hot) backups from Enterprise Manager that can back up, restore, and recover database files. This is a feature of the MS SQL 2000/2005 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 PLM 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

Copy the volumes from the source location to the destination device/location. After completing the backup, return Teamcenter to normal mode.

Use Virtual Device Interface (VDI)

Microsoft's Virtual Device Interface (VDI) is integral to all third-party SQL Server backup solutions, whether hardware or software. Introduced in SQL Server 7.0, 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 2000/2005, 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 2000/2005 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 math data 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.

Appendix

A Glossary

Appendix

A *Glossary*

A

Access Manager (AM)

Teamcenter application that enables the system administrator to grant users access to Teamcenter objects.

AM

See *Access Manager (AM)*.

assigned FSC

FMS server cache assigned as the volume or cache server for an FMS client cache. Each FMS client cache requires an assigned FSC to provide it with access to files. An assigned FSC is typically the FSC nearest to the client host. In small deployments, an assigned FSC can also serve as the parent FSC.

B

blobby volume

Alternate volume used to store data while backing up volumes and databases. The blobby volume provides continuous Teamcenter availability.

Business Modeler IDE

Teamcenter application that enables a customer to define data model objects such as business objects, classes, attributes, lists of values, and rules.

C

client

Role played by a software component of a system when it requests particular services be performed on its behalf by another entity, a server. See also *server*.

client tier

Teamcenter architectural tier that comprises the Teamcenter clients, Teamcenter integrations with third-party applications, and the third-party applications associated with the integrations.

corporate server

Host computer at the center of a Teamcenter network. This host contains the Teamcenter application root directory, Teamcenter data directory, licensing, File Management System (FMS), and volumes. For installations that include the Web tier (four-tier architecture), the corporate server also contains the Teamcenter server manager. Multiple application clients can map to or mount the corporate server.

D**data model**

Abstract model that describes how data is represented and used.

distribution server

See *rich client distribution server*.

E**enterprise archive (EAR)**

Enterprise application that requires a J2EE application server.

enterprise tier

Teamcenter architectural tier that comprises a configurable pool of Teamcenter C++ server processes and a server manager. Larger sites can distribute the pool of server processes across multiple hosts. Smaller sites can run the pool of servers on the same host as the Web tier.

F**FCC configuration file**

File that configures an individual FMS client cache (**fcc.xml**). The FCC configuration file defines such values as the parent FMS server cache location and the location and size of the client caches. Values defined in the FCC configuration file can override default values defined in the FSC configuration file.

File Management System (FMS)

System that manages uploading and downloading file data between clients and volumes in both two-tier and four-tier architecture deployments.

- FMS provides volume servers for file management, a shared server-level performance cache for shared data access between multiple users, a client-based private user cache for rich clients, and a transient datastore mechanism for transporting reports, PLM XML, and other nonvolume data between the enterprise and client tiers.
- FMS file caching enables placing the data close to the user, while maintaining a central file volume and database store.

FMS

See *File Management System (FMS)*.

FMS client cache (FCC)

FMS process that runs on a client host, uploading files to an FMS server cache process, requesting files from an FMS server cache process, and caching files on the client host. The FCC process manages two caches of whole files: a write cache containing files uploaded to a Teamcenter volume and a read cache containing files downloaded from a Teamcenter volume. It also manages one segment file cache for Teamcenter lifecycle visualization. Each Teamcenter rich client host requires a local FMS client cache.

FMS master configuration file

File that configures FMS (**fmssmaster.xml**). The FMS master configuration file describes the FMS network and defines groups of server caches. It can also define

default values for server caches and client caches, such as maximum sizes. Values defined in the server cache configuration file and in the client cache configuration file can override the default values defined in the master configuration file.

FMS server cache (FSC)

FMS process that runs on a server host and performs as a volume server (when running on a host where a volume is located or directly mounted) or a cache server (when running on a host where a volume is not located or directly mounted) and a configuration server. As a volume or cache server, the FSC checks all file access requests for a ticket that Teamcenter generates to authorize file access. As a cache server, it manages two segment file caches, one for downloading files and one for uploading files. As a configuration server, it provides FMS configuration information to file client caches and other FSCs. As a transient server, it delivers PLM XML and other transient files to clients. A minimum of one FSC must be deployed in any Teamcenter installation. Multiple FSCs can be deployed, with each FSC performing one designated purpose as either a volume, a cache, or a configuration server.

four-tier architecture

Teamcenter architecture that includes four tiers: resource tier, client tier, Web tier, and enterprise tier. Contrast with *two-tier architecture*.

four-tier deployment

Deployment of the Teamcenter four-tier architecture. The Web tier, enterprise tier, resource tier, and client tier can each be hosted on the same or separate computers.

FSC

See *FMS server cache (FSC)*.

FSC configuration file

File that configures an individual FMS server cache (**fsc.xml**). The FSC configuration file defines such values as the address of the master FSC, the maximum sizes of the segment file caches, and the upload time-out value. It can also define default values for FCCs and other FSCs.

FSC group

Group of server caches defined in the FMS master configuration file.

M**master FSC**

FMS server cache that reads the master configuration file directly from the FMS master host. An FSC is configured either to read the master configuration file directly from the master host or to download it from another FSC with access to it.

O**Oracle home**

Directory in which Oracle software is installed on the Oracle server node.

P**preference**

Configuration variable stored in a Teamcenter database and read when a Teamcenter session is initiated. Preferences allow administrators and users to configure many

aspects of a session, such as user logon names and the columns displayed by default in a properties table.

Q

QPL server

Quick part locator server. It provides a **qpl** daemon that can be used with DesignContext in the rich client. The **qpl** daemon coexists with all Teamcenter daemons. Without this daemon DesignContext does not work.

R

resource tier

Teamcenter architectural tier comprising the database server, database, file servers, and volumes.

rich client

Java-based user interface to Teamcenter installed on user workstations. The rich client accesses Teamcenter databases using a remote or local server. Compare to *thin client*.

rich client distribution server

Software that manages the connection between distribution server instances, the Web server, and the user's workstation when a rich client is deployed over the Web. The Over-the-Web Installer contacts the distribution server for the rich client files to download to the user's workstation.

S

server

System software component that performs a specifically defined set of software services on behalf of one or more clients. In a typical Teamcenter installation, servers are centralized on dedicated hosts that support a large number of clients. Clients are distributed on hosts connected to the servers via various networking techniques. See also *client*.

server manager

Process that manages a pool of Teamcenter server processes in a deployment of the four-tier architecture. The server manager starts and times out a configurable number of server processes to communicate with the Teamcenter database. A server assigner process assigns available server processes to user sessions. The server manager communicates with the Web tier application using either TCP or multicast protocol.

server pool

Pool of Teamcenter server processes running in the enterprise tier. A small deployment may have only one pool of server processes. For larger deployments, the pool of server processes is distributed as subpools across multiple hosts, with a server manager for each subpool. Server pools are applicable for deployments of the Teamcenter four-tier architecture only.

site

Individual installation of Teamcenter comprising a single Teamcenter database, all users accessing that database, and additional resources such as hardware,

networking capabilities, and third-party software applications (tools) required to implement Teamcenter at that site.

site ID

Unique identifier of a Teamcenter site. The site ID is used to generate internal identifiers for Teamcenter objects that must be unique throughout an enterprise. Once established, site IDs should not be modified.

site name

Unique name of a Teamcenter site stored in the database as a user-defined character string.

system administrator

Teamcenter user who is a member of the system administration group.

T**thin client**

Teamcenter user interface that provides a streamlined browser-based view of product information stored in a Teamcenter database. The thin client is configured in the Web tier, which creates and serves its Web pages to the client. Compare to *rich client*.

two-tier architecture

Teamcenter architecture that includes a resource tier and a client tier. The resource tier comprises the database server and database. The client tier comprises the Teamcenter rich client, third-party applications that integrate with the rich client, and a local server. This architecture supports only the Teamcenter rich client. Contrast with *four-tier architecture*.

two-tier deployment

Deployment of the Teamcenter two-tier architecture. In a typical deployment of the two-tier architecture, the rich client and its local server are installed on a user's workstation as are third-party applications that integrate with the rich client. The database server and the Teamcenter corporate server are installed on one or more separate computers.

V**volume**

Operating system directory controlled by Teamcenter and used to store the files managed by Teamcenter. When a user performs an action that causes Teamcenter to create a file, the file is created in the Teamcenter volume. Users cannot directly access the files in Teamcenter volumes; they must do so via a Teamcenter session.

W**Web Application Manager**

Graphical installation utility that generates supporting Web files (WAR and EAR format) for a named Web application. Web Application Manager also installs the rich client distribution server and creates distribution server instances.

Web tier

Teamcenter architectural tier that comprises a Java application running in a Java 2 Enterprise Edition (J2EE) application server. The Web tier is responsible for communication between the client tier and enterprise tier.

Index

A

Access multiple FMS databases with one FCC 8-115
Access remote sites via alias 8-122
ACQUIRE_REENTRANT_LOCK_
TIMEOUT 6-6
Action Manager daemon 2-1
Administering FCCs 8-49
Administering File Management
System 8-1
Administering FMS 8-45
Administering volumes 8-87
Administrative privileges 1-1
Administrative tasks 1-1
Alias access to remote sites 8-122
Allocating volume data 8-101–8-102
Allocating volume data using DTD file and
allocation rules 8-103
allowchunking 5-4
allowuntrustedcertificates 5-4
alwayspromptforusername 5-6
Application servers
Web tier 1-5
assignedfsc 8-55
assignedfsc element for FCC 8-56
assignment 8-55
assignment mode element for FCC 8-56
ASSIGNMENT_RETRY_LIMIT 6-7
ASSIGNMENT_RETRY_WAIT_
PERIOD 6-7
Audience 1-1
Audit logging
Audit points 8-59
Properties example 8-60
Audit logging configuration, FMS 8-59
Auto-start TCSS at Windows logon 9-22
Automatic server memory cleanup 9-2
Automatic server memory cleanup
disabled 9-4
Automatic server memory cleanup for
four-tier 9-3
Automatic server memory cleanup for
two-tier 9-3
Automatic server memory cleanup logs 9-4

Automatic server memory cleanup
settings 9-2

AUX_PATH system variable
configured for performance 9-25

B

Backing store file for preferences 9-10
Best practices for PKI authentication 8-75
Business logic server
Logging 10-3
Business logic server logging 6-29
Business Modeler IDE logging 10-26
deploy.log 10-26
Migration logs 10-26

C

cacerts.pem file 8-73
Certificate Signing Requests (CSRs) 8-72
Certificates imported into a keystore 8-72
Character set conversion 3-24
CIDR client maps 8-112
Client maps
CIDR 8-112
configuring 8-111
Domain name client maps 8-113
subnet/mask 8-111
Client tier
Four-tier 1-4
Two-tier 1-3
Client tier logging 10-8
Clients
Four-tier clients 1-4
Two-tier clients 1-3
Commands
db2cmd 3-4
db2look 3-3
db2move 3-3
db2start 3-1
db2stop 3-2
Compressing FMS files 8-117
Configure FMS audit logging 8-59
Configure PKI authentication 8-75
Configuring

FMS for HTTPS 8-68
Configuring business logic server
 logging 10-3
Configuring client maps 8-111
Configuring default local volumes 8-95
Configuring FMS for HTTPS 8-69
Configuring FMS to run multiple Teamcenter
 versions 8-67
Configuring FSC volume failover 8-99
Configuring server manager property
 files 6-3
Configuring Store and Forward
 behavior 8-95
Configuring TCCS 5-3
Configuring the FCC file warmer 9-21
Configuring volume failover during
 import 8-100
Connecting 3-2
connectiontimeout 5-4
Connectivity
 Teamcenter database 3-2
Container applications 5-9
Container TCCS applications 5-9
Control Center, opening 3-2
Conventions
 Syntax definitions 1-2
convert_license_log utility 4-3
Copying the FSC whole file cache 8-48
Copying volumes 8-104
Creating
 CSRs 8-72
 Key entry 8-70
Critical event monitoring 6-11
 Administrative interface 8-130
 fscMonitorConfig.xml sample code 8-128
 poolMonitorConfig.xml file 6-13
 poolMonitorConfig.xml sample
 code 6-15
Server manager 6-12
Server manager from the administrative
 interface 6-17
serverMonitorConfig.xml file 6-20
serverMonitorConfig.xml sample
 code 6-22
Teamcenter servers 6-19
Teamcenter servers from the administrative
 interface 6-23
Web tier 9-14
webtierMonitorConfig.xml file 9-15
webtierMonitorConfig.xml sample
 code 9-17

D

Database security 3-19, 3-30
Database service
 Starting 3-1
 Stopping 3-2
Database services 3-1, 3-8, 3-30
Databases
 Deletion 3-31
 Error logs 3-31
 Shutting down 3-2, 3-30
DB2 Control Center 3-2
DB2 Server
 Service shutdown 3-2
 Service startup 3-1
 System administration 3-1
db2cmd command 3-4
db2look command 3-3
db2move command 3-3
db2start command 3-1
db2stop command 3-2
Default automatic server memory cleanup
 settings 9-2
Default local volume best practices 8-95
Default local volume configuration 8-95
Default local volumes 8-88
 Multiple FSCs 8-94
 Side caching 8-94
 Single FSC 8-93
Default volumes 8-87
Deferred event times 2-1
directfsroute 8-55
directtransientvolume 8-56
Disabling server memory cleanup 9-4
Distribute network access load 8-105
DNS client maps 8-113
Domain name client maps 8-113

E

ENABLE_SERVER_HEARTBEAT 6-6
Encrypting passwords 2-5
Encryption key best practices 8-75
Encryption key stored in secure
 keystore 8-79
Enterprise tier 1-5
Enterprise tier logging 10-18
Environment variables
 LD_LIBRARY_PATH 3-6,
 3-15–3-16, 3-20
 LIBPATH 3-6, 3-15–3-16, 3-20
 NLS_LANG 3-24
 ORACLE_HOME 3-5–3-6
 ORACLE_SID 3-5–3-6
 PATH 3-6

SHLIB_PATH	3-6, 3-15–3-16, 3-20
TNS_ADMIN	3-28–3-29
Error codes	1-13
Error logs	3-31
Export/import	3-24
External load balancing	8-107
F	
Fast cache sizing	8-18
FCC	
Native	5-12
FCC assignment mode	8-56
FCC cache locked	8-50
FCC elements requiring restart	8-53
general	8-53
logging	8-54
segment cache	8-54
whole file cache	8-54
FCC file warmer	
filelist.txt file	9-20
filewarmer.properties file	9-20
logging	9-21
FCC file warmer configuration	9-21
FCC not responding to pipe attempts . . .	8-50
FCC offline	8-50
FCC reconfiguring	8-55
automatic	8-55
manual	8-56
FCC restart procedure	8-53
FCC running within TCCS	5-12
FCC runs within TCCS	8-51
FCC shut down	8-51
FCC will not start	8-50
FCC_CacheLocation	8-53
FCC_CachePurgeCycle	8-54
FCC_CacheTableHashSize	8-54
FCC_EnableDirectFSCRouting . . .	8-15, 8-55
FCC_FSCConnectionRetryInterval . . .	8-14, 8-55
FCC_HashBlockPages	8-54
FCC_IdleTimeoutMinutes	8-15
fcc.lck	8-54
FCC_LogFile	8-14, 8-54
FCC_LogLevel	8-14, 8-54
FCC_MaxExtentFiles	8-54
FCC_MaxExtentFileSizeMegabytes . . .	8-54
FCC_MaximumNumberOfFilePages . . .	8-54
FCC_MaximumNumberOfSegments . . .	8-54
FCC_MaximumReadCacheAge	8-54
FCC_MaximumWriteCacheAge	8-54
FCC_MaxReadCacheSize	8-54
FCC_MaxWANSources	8-14, 8-55
FCC_MaxWriteCacheSize	8-54
FCC_ProxyPipeName	8-14, 8-53
FCC_	
ReadCachePurgeSizePercentage	8-54
FCC_StatusFrequency	8-14, 8-53
FCC_TraceLevel	8-14, 8-54
FCC_TransientFileFSCSource . . .	8-15, 8-55
FCC_WebRaidThreshold	8-55
FCC_WholeFileCacheSubdirectories . .	8-54
FCC_	
WriteCachePurgeSizePercentage . . .	8-54
fcc.xml FMS file	8-5
FCCDefaults	8-55
FCCs, administering	8-49
FCCs, auditing	8-57
File import volume failover	8-100
File Management System	8-1
Benefits	8-1
Common caching system	8-2
Data distribution	8-1
Failure points	8-2
Multisite support	8-1
Pull-through caching	8-2
Segment file cache and delivery . .	8-2
Streamed data delivery	8-2
Supported network configuration . .	8-1
File Management System (FMS)	
logging	10-24
FSC_hostname(userID)_	
fsc.log.xml	10-25
FSC_hostname(userID)_	
plmconsole.txt	10-25
FSC_hostname(userID)_	
RTConsole.txt.xml	10-26
FSC_hostname(userID)_	
RTEvents.txt	10-25
FSC_hostname(userID)_	
RTEvents.txt.xml	10-26
FSC_hostname(userID)_	
RTRemotes.txt	10-25
FSC_hostname(userID)_	
RTRemotes.txt.xml	10-26
FSC_hostname(userID)_	
RTTraces.txt	10-25
FSC_hostname(userID)_	
RTTraces.txt.xml	10-26
\$FSC_ID.log	10-24
\$FSC_ID_startup.log	10-24
%FSC_ID%stderr.log	10-24
%FSC_ID%stdout.log	10-24
filelist.txt file	9-20
Files	
fmssmaster	8-59
filewarmer.properties file	9-20
FMS	

Compressing files for multisite transfer	8-117
Configured for HTTP and HTTPS	8-68
FMS benefits	8-1
FMS cache sizing	8-18
FMS client configuration file	8-13
FMS encryption key best practices	8-75
FMS encryption key stored in secure keystore	8-79
FMS event monitoring	8-124
fscMonitoringConfig.xml file	8-126
FMS files	
fcc.xml	8-5
fmssmaster.xml	8-5
fsc.xml	8-5
FMS for multiple Teamcenter versions	8-67
FMS master configuration file	8-6
FMS server configuration file	8-7
fms.ext	8-55
fms.hsh	8-54
fms.mf	8-55
fms.set	8-55
FMSClientCache component	
Initialize	5-12
fmssmaster file	8-59
fmssmaster.xml FMS file	8-5
Four-tier architecture	
Overview	1-4
Four-tier architecture performance	
Pool manager prerequisites	6-2
Four-tier architecture performance	
Tuning application servers	9-12
Four-tier architecture performance	
Using the .NET server manager interface	6-44
Four-tier architecture performance	
Using the server manager	6-1
Four-tier architecture performance	
Using the pool manager	6-1
Four-tier architecture performance	
Server manager prerequisites	6-2
Four-tier architecture performance	
Using the J2EE server manager interface	6-32
Four-tier architecture performance	9-1
Four-tier server memory cleanup	9-3
FSC client proxy in TcServer	8-73
FSC priority levels	8-99
FSC whole file cache	
Copying	8-48
FSCWholeFileCacheUtil utility	8-47
FSC whole file cache parameters	
FSC_DiskPercentFreeGoal	8-10
FSC_FilesPerWholeFileCacheDir	8-10
FSC_MaxCacheAgeDays	8-10
FSC_WholeFileCacheLocation	8-9
FSC_	
WholeFilePurgeInitialWaitMinutes	8-10
FSC_	
WholeFilePurgePeriodMinutes	8-10
FSC_DiskPercentFreeGoal	
parameter	8-10
FSC_FilesPerWholeFileCacheDir	
parameter	8-10
FSC_MaxCacheAgeDays parameter	
.	8-10
FSC_WebRaidThreshold	8-14
FSC_WholeFileCacheLocation	
parameter	8-9
FSC_WholeFilePurgeInitialWaitMinutes	
parameter	8-10
FSC_WholeFilePurgePeriodMinutes	
parameter	8-10
fsc.xml FMS file	8-5
fscread default example, audit logging	8-60
fscMonitorConfig.xml sample code	8-128
fscMonitoringConfig.xml file	8-126
FSCWholeFileCacheUtil utility	8-47
fwdproxy_cfg.properties file	5-6
tcpProxy.advanced.address_caching	5-8
tcpProxy.advanced.retry_delay	5-8
tcpProxy.connection.manual.all.port	5-7
tcpProxy.connection.manual.exceptions	5-7
tcpProxy.connection.manual.http.host	5-7
tcpProxy.connection.manual.http.port	5-7
tcpProxy.connection.manual.https.host	5-7
tcpProxy.connection.manual.https.port	5-7
tcpProxy.connection.manual.socks.host	5-7
tcpProxy.connection.manual.socks.port	5-7
tcpProxy.connection.ti	5-7
tcpProxy.connection.url	5-7
G	
Generating	
CSRs	8-72
Keystores	8-70
Getting started	1-1
Global pool properties	6-3
Global pool properties example	
settings	6-4
H	
Hardware load balancing	8-107
Health checks	8-108
HP Operations Manager	6-45
HTTP	

For FMS	8-68	LIBPATH	3-6, 3-15–3-16, 3-20
HTTP 400 error from health check . . .	8-108	License log files	4-3
HTTPS		License management	4-2
Certificates imported into a keystore	8-72	License usage reports	4-4
Configuration for FMS	8-69	license_warning_level preference	4-2
For FMS	8-68	LicenseUsedAuditTool tool	4-5
httpversion	5-4	Listener processes	3-6, 3-7, 3-9, 3-11–3-13, 3-15, 3-28
I		listener.ora file	3-7, 3-9, 3-13, 3-15, 3-28
Idle shutdown for TCCS	5-10	Load balancing FMS data	8-105
IDSM daemon	2-2	Load balancing using external hardware	8-107
Import volume failover	8-100	Load balancing volumes	8-105
Import/export	3-24	Lock file	5-10
Importing trusted certificates	8-72	Log files of server memory cleanup	9-4
Improve cache performance	8-132	Log Manager	10-1
init.ora file	3-25	log.properties file TCCS logging	5-13
Installation		LOG_VOLUME_LOCATION environment variable	
Over-the-Web	1-3	TCCS log files	5-13
Using TEM	1-3	log4j.xml file TCCS logging	5-14
Integrated backup modes	11-1	Logging	
J		Architecture	10-5
J2EE based server manager	1-5	Business logic server	6-29, 10-3
J2EE server manager administrative interface	6-32	Business logic servers	10-3
Java VisualVM	6-45	Client tier	10-8
JBossWeb logging	10-8	Enterprise tier	10-18
plmconsole.txt	10-9	Resource tier	10-24
plmconsole.txt.xml	10-10	Server manager	6-27–6-28
RTConsole.txt	10-9	Translations	10-27
RTConsole.txt.xml	10-10	Web tier	10-8
RTEvents.txt	10-9	Logging for TCCS	5-13
RTEvents.txt.xml	10-10	Logging levels for TCCS	5-14
RTRemotes.txt	10-9	Logging the syslog file	10-3
RTRemotes.txt.xml	10-10		
RTTraces.txt	10-9		
RTTraces.txt.xml	10-10		
WebTier.log	10-8		
WebTier.log.xml	10-9		
JConsole	6-45		
K			
kerberosconfig	5-6	make_user utility	4-1
key entry	8-70	Managing client maps	8-111
keystore	5-4, 8-70	Managing FCCs	8-49, 8-57
keystorepassword	5-4	Managing server memory cleanup	9-2
Kill command to stop FCC	8-51	Managing volumes	8-87
krb5path value	5-6	MAX_POOL_PROCESSING_	
		INTERVAL	6-6
L		maxconnectionsperhost	5-4
LD_LIBRARY_PATH	3-6, 3-15–3-16, 3-20	maxidletime	8-15
		maxretriesreverseproxy	5-4
		Memory mapped files	9-9
		Metadata shared memory	9-7
		MIN_POOL_PROCESSING_	
		INTERVAL	6-6
		Mixed mode	3-30

Monitoring critical events	6-11	Net-Library	3-31
Administrative interface	8-130	NLS_LANG environment variable	3-24
fscMonitorConfig.xml sample code .	8-128		
poolMonitorConfig.xml file	6-13		
poolMonitorConfig.xml sample			
code	6-15	o	
Server manager	6-12	ODBC	3-31
Server manager from the administrative		ODS daemon	2-2
interface	6-17	ORA_DBA group	3-19
serverMonitorConfig.xml file	6-20	Oracle	
serverMonitorConfig.xml sample		Character-set conversion	3-24
code	6-22	Export/import	3-24
Teamcenter servers	6-19	init.ora file	3-25
Teamcenter servers from the administrative		Initialization files	3-25
interface	6-23	Listener processes	3-6–3-7,
Web tier	9-14	3-9, 3-11–3-13, 3-15, 3-28	
webtierMonitorConfig.xml file	9-15	NLS_LANG environment variable	3-24
webtierMonitorConfig.xml sample		Parameter files	3-25
code	9-17	Service parameter file	3-25
Monitoring FMS events	8-124	spfile.ora file	3-25
fscMonitoringConfig.xml file	8-126	System administration	3-4
Moving volumes within an		Oracle logging	10-15
enterprise	8-104	plmconsole.txt	10-16
Multi-Site logging	10-22	plmconsole.txt.xml	10-17
idsmID.jnl	10-23	RTConsole.txt	10-16
idsmID.log	10-23	RTConsole.txt.xml	10-17
idsmID.syslog	10-23	RTEvents.txt	10-16
odsID.jnl	10-24	RTEvents.txt.xml	10-17
odsID.log	10-23	RTRemotes.txt	10-16
odsID.syslog	10-23	RTRemotes.txt.xml	10-18
Multiple FMS databases accessed with one		RTTraces.txt	10-17
FCC	8-115	RTTraces.txt.xml	10-18
Multiple TCCS environments	5-8	WebTier.log	10-16
Multiple Teamcenter versions on one		WebTier.log.xml	10-17
system	8-67	Oracle Net Assistant	3-29
Multisite routing FSCs	8-120	Oracle Recovery Manager	
Multisite transfer of compressed		Benefits	11-4
files	8-117	Features	11-3
Multiuser support for Windows	8-67	ORACLE_HOME	3-5–3-6
N		ORACLE_SID	3-5–3-6
Native authentication	3-19	OracleTNSListener process	3-6,
Native FCC	5-12	3-11–3-12, 3-27–3-28	
Native FSC client proxy in TcServer . . .	8-73		
.NET based server manager	1-5		
.NET logging without Log Manager . . .	10-18		
.NET server manager administrative			
interface	6-44		
History log	6-44	P	
Stop servers	6-43	parentfsc	8-55
.NET Web tier		Password encryption	2-5
Server manager	1-5	PATH	3-6
Windows	1-5	PATH system variable	
		configured for performance	9-25
		Performance of four-tier architecture . .	9-1
		Performance of rich client	9-18
		Pipe attempts, FCC not responding . .	8-50
		PKI authentication	8-75
		PKI authentication best practices . .	8-75

PLM XML logging	10-22	run_tc_ods script	2-2
plxml_log_timestamp.log	10-22		
xml-file-name.log	10-22		
Pool manager .NET administrative interface	6-44		
Pool manager J2EE administrative interface	6-32		
Pool manager prerequisites	6-2		
Pool manager, starting	6-2		
poolMonitorConfig.xml file	6-13		
poolMonitorConfig.xml sample code	6-15		
Prepopulating FSCs	8-132		
Prerequisites	1-1		
Priority levels of FSCs	8-99		
Process logs	10-2		
PROCESS_CREATION_DELAY	6-5		
PROCESS_CREATION_DELAY setting	6-7		
PROCESS_MAX	6-5		
PROCESS_MAX setting	6-8		
PROCESS_READY_TIMEOUT	6-6		
PROCESS_TARGET	6-5		
PROCESS_TARGET setting	6-8		
PROCESS_WARM	6-5		
PROCESS_WARM setting	6-9		
Processing events	2-1		
R			
Reallocation rules XML file	8-103		
Reassigning files to different volumes	8-101–8-102		
Reassigning files to different volumes using DTD file	8-103		
-reconfig argument in fccstat utility	8-56		
Reconfiguring an FCC	8-55		
automatic	8-55		
manual	8-56		
Reconfiguring FCCs	8-49		
Reset user environment	8-54		
Resource tier			
Four-tier	1-5		
Two-tier	1-3		
Resource tier logging	10-24		
Restart an FCC, procedure	8-53		
Restarting FCCs	8-49		
Restarting TCCS	5-10		
reverseproxy_config.xml file	5-8		
Rich client			
Description	1-12		
Overview	1-12		
Rich client startup performance	9-18		
Routing FSCs between sites	8-120		
run_tc_idsm script	2-2		
S			
Secure fscadmin commands	8-76		
Secure keystore for FMS encryption key	8-79		
Secure the FMS encryption key	8-79		
Security			
Databases	3-19		
Native authentication	3-19		
Semaphores	3-22		
Server logging	6-29		
Server manager			
Configuring property files	6-3		
J2EE based	1-5		
Logging	6-27		
.NET based	1-5		
Server manager administrative interface for .NET	6-32, 6-44		
Server manager logging	6-28, 10-18		
administration.log	10-21		
audit.log	10-22		
\$FSC_ID_startup.log	10-22		
%FSC_ID%stderr.log	10-22		
%FSC_ID%stdout.log	10-22		
install.log	10-20		
pomutilities.log	10-21		
security.log	10-20		
ServerManager.log	10-18		
ServerManager.log.xml	10-19		
system.log	10-21		
tcserverpid.jnl	10-19		
tcserverpid.log	10-19		
tcserverpid.orblog	10-20		
tcserverpid.syslog	10-19		
Server manager prerequisites	6-2		
Server manager, starting	6-2		
Server memory cleanup	9-2		
Server memory cleanup default settings	9-2		
Server memory cleanup disabled	9-4		
Server memory cleanup for four-tier	9-3		
Server memory cleanup for two-tier	9-3		
Server memory cleanup logs	9-4		
SERVER_HEARTBEAT_INTERVAL	6-6		
SERVER_HOST	6-7		
SERVER_PARAMETERS	6-7		
SERVER_RETRY_LIMIT	6-6		
SERVER_RETRY_WAIT_PERIOD	6-6		
serverMonitorConfig.xml file	6-20		
serverMonitorConfig.xml sample code	6-22		

Setting PROCESS_CREATION_DELAY	6-7	Multiple FSCs	8-94
Setting PROCESS_MAX	6-8	Single FSC	8-93
Setting PROCESS_TARGET	6-8	Subnet/mask client maps	8-111
Setting PROCESS_WARM	6-9	Subscription daemon	2-3
Shared key security	8-75	Subscription events	2-3
Shared memory		Subscription Manager daemon	2-3
Metadata	9-7	SVR4 platforms	3-18
Text Server data	9-9	Syntax definition conventions	1-2
Shared memory for preferences	9-10	syslog logging	10-3
Shared network	8-122	System Administration Guide	8-73
Sharing a single TcServer instance	9-12	System administration, Oracle	3-4
SHLIB_PATH	3-6, 3-15–3-16, 3-20		
Shutting down an FCC	8-51		
Side caching with default local volumes	8-94		
Single file recovery			
Object model	11-7	Task daemon	2-4
Rich client interface	11-8	Task logs	10-2
Sizing the FMS cache	8-18	Task Manager daemon	2-4
sockettimeout	5-5	Tasks	1-1
Software load balancing	8-105	TCCS	5-9
spfile.ora file	3-25	Allow chunking	5-4
SQL logging	6-31	allowuntrustedcertificates	5-4
SQL Net		alwayspromptforusername	5-6
Net-Library settings	3-31	Auto-start at Windows logon	9-22
ODBC	3-31	connectiontimeout	5-4
TCP/IP	3-31	FCC	5-1, 5-12
SQL Server		fwdproxy_cfg.properties file	5-6
Database deletion	3-31	httpversion	5-4
Database security	3-30	kerberosconfig	5-6
Error logs	3-31	keystore	5-4
Net-Library settings	3-31	keystorepassword	5-4
ODBC	3-31	krb5path value	5-6
Service shutdown	3-30	log.properties file	5-13
Service startup	3-30	log4j.xml file	5-14
System administration	3-29	Logging	5-13
TCP/IP	3-31	Logging levels	5-14
sqlnet.ora file	3-19, 3-28–3-29	maxconnectionsperhost	5-4
stalechecking	5-5	maxretriesreverseproxy	5-4
Starting database services	3-1	Multiple environments	5-8
Startup performance of rich client	9-18	reverseproxy_config.xml file	5-8
Stopping an FCC	8-51	sockettimeout	5-5
Stopping FCCs	8-49	stalechecking	5-5
Store and Forward best practices	8-95	tccs.xml file	5-3
Store and Forward configuration	8-95	TcMEM	5-1, 5-12
Store and forward files	8-88	tcpProxy.connection.type	5-6
Store and Forward volume with multiple FSCs	8-94	TcServerProxy	5-1, 5-10
Store and Forward volume with one FSC	8-93	totalmaxconnections	5-5
Store and forward volumes		truststore	5-5
Side caching	8-94	truststorepassword	5-5
Store and Forward volumes		Unsupported clients	5-12
		usesinglecookieheader	5-5
		TCCS configuration files	5-3
		TCCS container applications	5-9
		TCCS contains FCC	8-51
		TCCS idle shut down	5-10

TCCS lock file	5-10
TCCS log files	
LOG_VOLUME_LOCATION environment variable	5-13
TCCS restart	5-10
tccs.xml file	5-3
TcMEM	5-12
TCP/IP	3-31
tcpProxy.advanced.address_caching	5-8
tcpProxy.advanced.retry_delay	5-8
tcpProxy.connection.manual.all.host	5-7
tcpProxy.connection.manual.all.port	5-7
tcpProxy.connection.manual.exceptions	5-7
tcpProxy.connection.manual.http.host	5-7
tcpProxy.connection.manual.http.port	5-7
tcpProxy.connection.manual.https.host	5-7
tcpProxy.connection.manual.https.port	5-7
tcpProxy.connection.manual.socks.host	5-7
tcpProxy.connection.manual.socks.port	5-7
tcpProxy.connection.type	5-6
tcpProxy.connection.url	5-7
TcProxyClient	5-10
TcServer instance	
Shared	9-12
TcServerProxy	5-10
Teamcenter database	3-2
Teamcenter server journaling	6-31
Teamcenter server logging	6-29
Thin client	
Description	1-12
THREAD_POOL_INVOKING_SERVERS	6-6
TIE/PIE logging	
TcWebTier.evtx	10-18
TNS_ADMIN environment	
variable	3-28–3-29
tnslsnr process	3-6, 3-12–3-13, 3-15, 3-27–3-28
tnsnames.ora file	3-28–3-29
totalmaxconnections	5-5
Translation server logging	10-27
Adminclient.log	10-28
AdminClient.log.xml	10-29
History.log	10-28
History.log.xml	10-29
Module_ID.log	10-27
Module_ID.log.xml	10-29
Scheduler.log	10-27
Scheduler.log.xml	10-29
taskID_m.log	10-27
taskID_m.log.xml	10-28
taskID_s.log	10-27
taskID_s.log.xml	10-28
taskID_ts.log	10-27
taskID_ts.log.xml	10-29
Translations logging	10-27
Troubleshooting	
SQL Server	3-31
Trusted certificates imported into a keystore	8-72
truststore	5-5
truststorepassword	5-5
Tuning four-tier architecture	9-1
Two-tier architecture	
Deployment	1-4
Overview	1-3
Two-tier server memory cleanup	9-3
U	
UNIX semaphores	3-22
Used kill command to stop FCC	8-51
User environment reset	8-54
user-name_TcRAC.log	10-8
usesinglecookieheader	5-5
Using default volumes	8-87
Using the pool manager	6-1
Using the server manager	6-1
Using this guide	1-1
Using volumes	8-87
Utilities	
make_user	4-1
V	
Virtual Device Interface (VDI)	11-10
Volume allocation rules XML file	8-103
Volume data allocation	8-101–8-102
Volume data allocation DTD file	8-103
Volume failover	8-99–8-100
Volumes	8-87
Configuring FSC failover	8-99
Configuring volume failover at import	8-100
Failover at import	8-100
FSC Failover	8-99
Load balancing	8-105
Load balancing with external hardware	8-107
Moving within an enterprise	8-104
Volumes for store and forward	8-88
W	
Web tier	
Administrative interface	9-13

FMS administrative interface	8-131	RTTraces.txt	10-14
J2EE	1-4	RTTraces.txt.xml	10-15
Web tier logging	10-8	WebTier.log	10-13
WebLogic logging	10-10	WebTier.log.xml	10-14
plmconsole.txt	10-11	webtierMonitorConfig.xml file	9-15
plmconsole.txt.xml	10-12	webtierMonitorConfig.xml sample code	9-17
RTConsole.txt	10-11	Whole file cache	
RTConsole.txt.xml	10-12	Copying	8-48
RTEvents.txt	10-11	FSCWholeFileCacheUtil utility	8-47
RTEvents.txt.xml	10-12	Whole file cache parameters	
RTRemotes.txt	10-12	FSC_DiskPercentFreeGoal	8-10
RTRemotes.txt.xml	10-13	FSC_FilesPerWholeFileCacheDir	8-10
RTTraces.txt	10-12	FSC_MaxCacheAgeDays	8-10
RTTraces.txt.xml	10-13	FSC_WholeFileCacheLocation	8-9
WebTier.log	10-11	FSC_	
WebTier.log.xml	10-12	WholeFilePurgeInitialWaitMinutes	8-10
WebSEAL health check	8-108	FSC_	
WebSphere logging	10-13	WholeFilePurgePeriodMinutes	8-10
plmconsole.txt	10-13	Windows authentication mode	3-30
plmconsole.txt.xml	10-15	Working with volumes	8-87
RTConsole.txt	10-14		
RTConsole.txt.xml	10-15		
RTEvents.txt	10-14		
RTEvents.txt.xml	10-15		
RTRemotes.txt	10-14		
RTRemotes.txt.xml	10-15		

X

XML for volume allocation rules	8-103
---	-------