Teamcenter 8.3

Multi-Site Collaboration Guide

Publication Number
PLM00028 F

Proprietary and restricted rights notice

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

© 2010 Siemens Product Lifecycle Management Software Inc. All Rights Reserved.

All trademarks belong to their respective holders.

Contents

Figures

Part I Introduction

Chapter

1 Getting started

Chapter

1 Getting started

You can use Multi-Site Collaboration to easily share product information across your entire enterprise. Multi-Site Collaboration allows the exchange of Teamcenter data objects between databases. Each database should be easily accessible using TCP/IP, either over the Internet or your company intranet.

You must coordinate configuration of Multi-Site Collaboration with the system administrators of the other Teamcenter databases that are participating in your Multi-Site Collaboration environment. Information about all participating Teamcenter database sites must be stored in each database and in the site preference files. In addition, you must identify the network nodes you want to run Multi-Site Collaboration server processes for these databases and configure those systems to run the processes.

Before you begin

Prerequisites	You do not need any special permissions to use Multi-Site Collaboration. You must have system administration and database administration privileges to configure Multi-Site Collaboration.
Enable Multi-Site Collaboration	As system administrator, you enable Multi-Site Collaboration by setting it up and configuring it. The setup and configuration required is determined by what data you intend to share and how you intend to synchronize the data. Once Multi-Site Collaboration is setup and configured, no further action is required for you to use it.
Configure Multi-Site Collaboration	The tasks required to configure Multi-Site Collaboration depend on the how the sites that participate are connected, how data is coupled between sites, and other considerations. These are determined during planning and setup of your Multi-Site Collaboration network. For information about planning and configuring Multi-Site Collaboration, see *Configuring and administering Multi-Site Collaboration*.
Start Multi-Site Collaboration	Multi-Site Collaboration is accessible from within the thin client or rich client interface.

Chapter

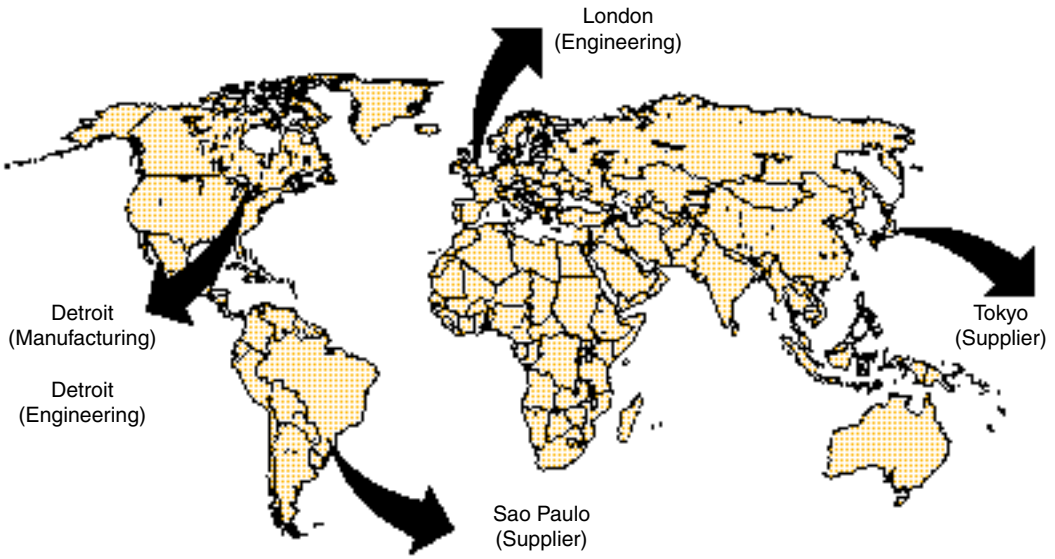
2 Basic concepts

Chapter

2 *Basic concepts*

To clearly understand the issues involved with sharing product information across an entire enterprise, consider how the XYZ Widget Corporation shares data without the benefit of Multi-Site Collaboration.

The following figure shows that the XYZ Widget Corporation has engineering sites in Detroit and London, a manufacturing site in Detroit, and suppliers in Tokyo and São Paulo. Each of these sites currently stores their product information in separate databases.



Multiple sites

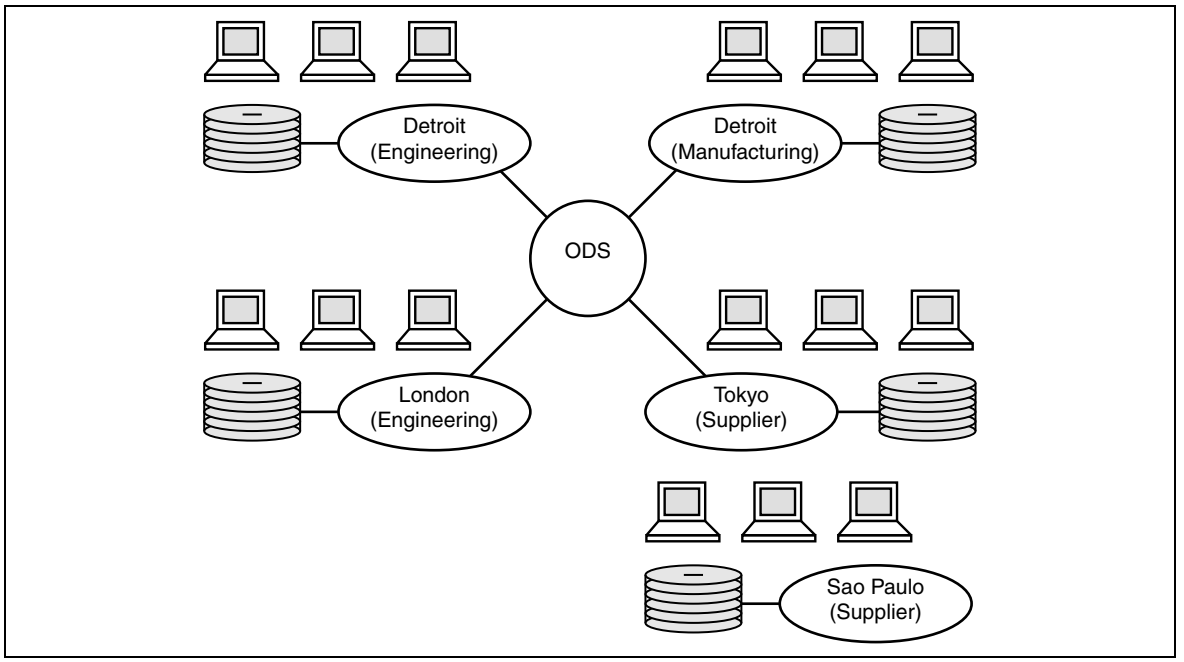
During product development, the engineering sites in Detroit and London occasionally share small amounts of data with one another and with their suppliers in São Paulo and Tokyo. This is accomplished by manually exporting product information as objects, transferring these objects using File Transfer Protocol (FTP) or removable media (DAT) to the desired site, and manually importing them into the databases.

After product development completes, engineering data is manually exported and transferred to the Detroit manufacturing site and imported into that database.

Although this solution can work acceptably on a limited basis, it requires too much labor and too many ad hoc arrangements to be viable for routinely sharing large amounts of product information across this enterprise.

Multi-Site Collaboration solution

The Multi-Site Collaboration solution provides semi automated real-time data sharing across the entire enterprise. It automates many of the operations that had to be performed manually in our first example.



Unconnected sites

Practical example	XYZ Widgets decides to link both the Detroit sites with the London and Tokyo sites using a high-speed wide area network (WAN). They also decide that the supplier in São Paulo would not be sharing enough product information with the other sites to justify a WAN connection.
Unconnected sites	The São Paulo site is not connected to the other sites through a local or wide area network (LAN or WAN). Data sharing with São Paulo must be accomplished using manual export, transfer, and import as described in our first example. However, because the XYZ Widget Corporation has implemented a Multi-Site Collaboration network, some tracking of objects in the São Paulo database must be performed for the benefit of the other sites.
ODS site	The Multi-Site Collaboration solution uses a special site called an Object Directory Services (ODS) site. The ODS site maintains a record of each object in the entire Multi-Site Collaboration network. The ODS does not store the objects, it maintains a record that is similar to a library card; it tells you which site is currently storing it and some basic information about it (enough information so you can decide if it is the object you are looking for).

Sites, facilities, and the Multi-Site Collaboration network

Three very common terms have very specific meanings in Multi-Site Collaboration: sites, facilities, and network. These are defined in the following table.

Term	Definition
Site	A single database and all users that access the database. Additionally, includes any non-Teamcenter resources such as hardware, networking capabilities, and third-party software applications (tools) required to implement Teamcenter at that site.
Facility	A physical location (for example, manufacturing plant, design center, and so forth) in your enterprise. Do not to confuse sites and facilities. Sites are databases; facilities are buildings. One facility can have multiple sites.
Network	A federation of independent sites that share data within the same enterprise. Though each site is independent, it is able to operate on and share data within the Multi-Site Collaboration network. Multi-Site Collaboration intentionally imposes as few restrictions and limitations on autonomous site activity as possible.

Data replication

Data replication, through import and export functions, is the foundation of Multi-Site Collaboration. In contrast, most other distributed solutions simply export a copy of an object into a remote applications memory and either discard it when the application exits, or save it by reimporting the new version back into the original database.

The latter approach has the advantage of using less disk space because each object has only one disk copy in the entire network. However, it results in poor performance because the object must be transmitted over the network every time a remote user wants to access it.

The data replication approach used by Multi-Site Collaboration does use more disk space because objects are replicated at various sites. However, after an object is copied to another site, access is as fast as any other object in the local database.

A replication-based distributed solution must address the following considerations:

Object	Consideration
Data integrity	As an object is replicated to various sites, how do you determine which object is the latest version of an object? This is especially true if users are allowed to modify replicated objects.
Security	Without proper security controls, replicated product information could fall into the hands of people not authorized to have it.

Object	Consideration
Auditing and tracking	A replication-based system must provide some method of tracking all replicas of an object not only for audit purposes, but also for ensuring that all replicas are updated when the original is modified.
Rules	Multi-Site Collaboration addresses these considerations by imposing the following rules on object replication: • Only the master object can be replicated. You cannot replicate *replicas*. When an object is initially created and saved in a database, that instance is considered the master object until such time as it is exported with transfer of ownership. **Note** An exception to this rule is the Multi-Site Collaboration hub configuration. • Only the master object can be modified. All replicas of the master object are read-only. This ensures that the master object is always the latest copy. • When you export an object, you must specify which sites are authorized to import it. This ensures that no unauthorized replicas are made and stores tracking information with the master object. • When transferring ownership to another site, only one site can be specified. This ensures that there is only one master object in the network. • After it is replicated, a master object cannot be deleted until all replicas are deleted. This ensures network-wide referential integrity.

Synchronization

A replication-based solution must ensure that replicas are kept up-to-date when the master object is modified. Multi-Site Collaboration addresses this by maintaining export records and providing synchronization facilities.

Item	Description
Export records	When an object is exported, export records are created for each target site specified. Each export record contains the site ID of each target site and the date of the last export to that site. Export records are always associated (and stored) with the master object. For items, a special Item Export record is also created to record the import/export options used so that these same options can be used to synchronize the item.
data_sync utility	When you modify a master object, use this utility to update any replicas. You must have system administrator privileges to use this utility. The process of keeping replicated data up-to-date is called synchronization. Optionally, synchronization may be limited to visualization data that is directly or indirectly related to datasets. For information about this utility, see the *Utilities Reference*.
sync_on_demand utility	You may update replicated objects as they require using this utility. You can select a component, assembly, or object for a synchronization report that allows you to determine if synchronization is required and to select the specific components to synchronize. For information about this utility, see the *Utilities Reference*.
Automatic synchronization	The end user who replicates an object may specify that the replica be synchronized automatically when the master object is modified. The replica is then synchronized automatically using the Multi-Site Collaboration automatic synchronization functionality.

Publishing and unpublishing

Participating sites in a distributed network must have a reliable way of controlling which data they want to share with the rest of the network. With Multi-Site Collaboration, you can publish and unpublish objects either singly or in a batch.

Item	Description
Publishing	Publishing an object makes that object available to other sites. When you publish an object, a publication record is created in the ODS that can be read and searched by other sites. Until you publish an object, it can only be seen by the local owning site; other sites are not aware that it exists.
Unpublishing	Unpublishing an object reverses the procedure. The object is accessible only by the local owning site.

Item	Description
data_share utility	Publish or unpublish objects in a batch using this utility. For additional information, see the *Utilities Reference*.

Object ownership and protection

In a normal (for example, nondistributed) environment, the ownership and protection of objects is straightforward and generally transparent to users. However, in a distributed environment, the level of complexity is greatly increased in order to extend object protection across an entire network.

In addition to the familiar concepts of owning user and owning group, Multi-Site Collaboration uses the concept of site ownership. The owning site is the site where the master object resides. It is the only site where the object can be modified or where you can obtain a replicated copy of the master object.

The owning site is a property of any object, and the owning site can be found using the **Properties** dialog box.

When an object is replicated by a remote site, the owning site property goes along with it. However, other aspects of access control may vary for each replica according to the environment of the replicating (that is, remote) site. The following describes access control on replicas:

1. All replicas are read-only objects, regardless of whether the site uses rules-based or object-based protection.

2. When an object is replicated, the owning user and owning group for the replica are determined as follows:

   - If the owning user and owning group of a Master object are both defined at the importing site, the imported copy (replica) is owned by this user and group following the import. The ownership is fully preserved.

   - If either the owning user or owning group of a Master object is not defined at the importing site, the imported copy (replica) is owned by the user performing the import; the owning group is that user's current group at the time of the import.

   - If the **TC_retain_group_on_import** site preference is defined and set to **TRUE**, and the owning group is defined at the importing site, the original owning group is preserved.

     Note

     These rules are also true when site ownership is transferred from one site to another.

     Caution

     If the group set in this preference is not defined at the importing site, this preference has no effect and the group is set to the default group of the user doing the import.

3. When an object is exported from a site using traditional object-based protection (that is, not using rules-based protection) and imported into a site using rules-based object protection, access controls at the importing site apply (subject to the limitation that remote objects are always read-only). This is true regardless of whether site ownership is transferred or not.

   - Site autonomy permitted: Multi-Site Collaboration imposes as few restrictions and limitations on autonomous site activity as possible. This includes object protection and ownership. Sites are not required to define users from other sites in their database, and each site is free to choose the object protection scheme (object-based or rules-based) used at their site. Furthermore, if rules-based object protection is used, each site is free to define the rules in effect at their site.

   - Site unity recommended: Siemens PLM Software recommends that all sites use rules-based object protection and define similar rules so that access to shared objects is uniform across the entire Multi-Site Collaboration network. Defining a consistent set of users for all sites is recommended whenever possible.

Chapter

3 Version interoperability

Chapter

3 Version interoperability

When a Multi-Site Collaboration site is upgraded to a new version, it is not necessary to upgrade all other sites in the Multi-Site Collaboration network at the same time. When a new major version is released, it is interoperable with all sites running earlier versions as long as the difference in major version numbers is not more than 2.

Although interoperability is guaranteed, there can be some limitations. For example, transfer of ownership of certain types of objects from a higher release version to a lower one may not be allowed. In most cases, new features introduced in a new release are not available when communicating with a remote site running an earlier version. The version of the server dictates what the client can do.

Teamcenter 8.3 provides 128 bytes for item IDs and names. Engineering Process Management sites provide only 32 bytes for these attributes and earlier versions of Teamcenter can be configured for either length attributes. In a Multi-Site Collaboration environment, a warning displays when a 128-byte site sends an item ID or name longer than 32 bytes to a 32-byte site. Therefore, Siemens PLM Software recommends that you upgrade Engineering Process Management sites in your Multi-Site environment to a Teamcenter site that to supports 128-byte item IDs and names

Teamcenter 8.2 sites can be multilingual sites. Multingual sites provide localized attribute values for certain attributes in exported data and can accept localized attribute values in imported data. There are certain conditions for transfers between monolingual and multilingual sites of which you must be aware.

For information about these conditions, see *Supporting multiple languages*.

There are some data model differences between some earlier versions of Teamcenter and Teamcenter 8.3. Multi-Site Collaboration support exchange of data between the versions by mapping the data during the export and import actions. The class subtypes from earlier versions are mapped to subclasses in Teamcenter 8.3. The subclasses are mapped back to subtypes when the data is transferred from Teamcenter 8.3 to an earlier Teamcenter version.

For more information, see *Compatibility with earlier versions*.

Siemens PLM Software supports Multi-Site Collaboration interoperability between versions. Teamcenter 8.3 interoperates with Teamcenter 2007, Teamcenter Engineering 2005, Teamcenter Engineering 2007, and Teamcenter's engineering process management 2008 as shown in the following table.

Source (any version of base)	Target
Teamcenter 2007	Teamcenter 8.1 or later

Source (any version of base)	Target
Teamcenter Engineering 2005 SR1	Teamcenter 8.1 or later
Teamcenter Engineering 2007	Teamcenter 8.1 or later
Teamcenter's engineering process management 2008	Teamcenter 8.1 or later

Certain functional limitations, configuration requirements, and schema changes are inherent to interoperability because the data model evolves to provide increased functionality. In most cases, schema differences are handled transparently by Multi-Site Collaboration. However, some schema changes require some Teamcenter 8.3 features to be temporarily disabled until all sites are upgraded to Teamcenter 8.

There are two categories of changes:

- *General changes* apply to all data sharing scenarios
- *Application-specific changes* apply only to specific classes or applications. Application-specific changes may not be relevant to your installation.

The following are previously documented restrictions:

- Instances of classes introduced in Teamcenter 8.3 are not imported into previous versions; they are ignored.
- New attributes in Teamcenter 8.3 added to POM classes that existed in earlier versions are exported from Teamcenter 8.3 but are not imported into earlier versions. When you import these attributes from earlier versions to Teamcenter 8.3, they are assigned null values.
- If a new type is added in Teamcenter 8.3, such as a relation type or dataset type, and is exported to earlier versions, the type must be defined at the earlier version site using an appropriate tool. An appropriate tool may be Business Modeler IDE for a Teamcenter site or the **install_types** utility in Engineering Process Management. For relation types, you can define the **ITEM_do_not_export_relation_if_type_excluded** site preference at the Teamcenter 8.3 site to list the new relation types added in Teamcenter 8.3 that should not be exported to the earlier version.

The following are general requirements applicable in Teamcenter 8.3:

- If an ODS is shared by Teamcenter 8.3 and earlier versions, the ODS site must first be upgraded to Teamcenter 8.3.

  Teamcenter 8.3 clients cannot publish to an ODS server running an earlier version.
- Teamcenter 8.3 sites that must share data with any Engineering Process Management versions (Teamcenter Engineering 2005 SR1, Teamcenter Engineering 2007, and Teamcenter's engineering process management 2008), cannot use longer IDs and names. Verify that the **TC_Allow_Longer_ID_Name** site preference is either not defined or set to **FALSE**. You can set this preference to **TRUE** when all participating sites are at Teamcenter 2007 or later.

- Teamcenter 8.3 sites that must share data with earlier versions must temporarily disable the sequence feature that was introduced in Teamcenter 2007, by setting the **TCCheckoutReserveOnly** site preference to **WorkspaceObject**. By default, the sequence feature is for all subclasses of **WorkspaceObject**. You can disable or modify this preference for specific classes once all participating sites are at Teamcenter 2007 or later.

- For Teamcenter 8.3 and sites that predate Teamcenter 8.3, set the **TC_ALLOW_INCOMPATIBLE_TYPES** environment variable to **TRUE**.

- Several allocation schema changes are made in Teamcenter 2007. To exchange allocations data, sites that predate Teamcenter 2007 and that interoperate with Teamcenter 8.3 must run the **upgrade_data_v1001mpx_multisite.default** script on the version that predates Teamcenter 2007.

  The **upgrade_data_v1001mpx_multisite.default** script, which is available with Teamcenter Engineering 2005 SR1 MP3 and subsequent versions, makes the following changes to the allocations schema:

  – Modifies the properties of **Allocation** attributes **source_absOcc_tag**, **target_absOcc_tag**, and **map_rev_tag** to allow null values.

  – Modifies the properties of **AllocationMap** attributes **source_bv_tag** and **target_bv_tag** to allow null values.

  – Introduces the **ManagedRelation** class, a subclass of **WorkspaceObject**.

- For Teamcenter Engineering 2005 SR1, Teamcenter Engineering 2007, and Teamcenter's engineering process management 8 sites to interoperate with Teamcenter 8.3 sites, the administrator must make the following schema change manually at the Engineering Process Management site:

  – Make valid the **is_frozen** attribute from **POM_object** with **NULL**:

```
install -mod_attr infodba infodba dba ItemRevision
   variant_expression_block POM_attr_follow_on_export +
install -mod_attr infodba infodba dba ItemRevision
   variant_expression_block POM_null_is_valid +
install -mod_attr infodba infodba dba PSOccurrence
   variant_condition POM_attr_follow_on_export +
install -set_attr infodba infodba dba PSOccurrence
   variant_condition NULLTAG
install -mod_attr infodba infodba dba PSOccurrence
   variant_condition POM_null_is_valid –
install -gen_xmit_file -u=infodba -p=infodba -g=dba
```

- For all Teamcenter 2007 sites that must interoperate with Teamcenter 8.3 sites, the administrator must make the following schema changes manually at the Teamcenter 2007 site:

  – Make valid the **resource_tag** attribute from **ScheduleMember** with **NULL**.

```
install -mod_attr -u=infodba -p=infodba -g=dba
   ScheduleMember resource_tag POM_null_is_valid +
```

  – Install **gen_xmit_file**, which is required after the schema changes.

```
install -gen_xmit_file -u=infodba -p=infodba -g=dba
```

Note

The administrator must regenerate the POM schema and transmit files after making these schema changes. The new transmit files must be distributed to the other sites.

- A remote checked-out assembly cannot be saved.

  When using Teamcenter Integration for NX assemblies in a Multi-Site Teamcenter configuration, you cannot save a remote checked-out assembly.

  Use Multi-Site Collaboration import/export with transfer ownership instead of checkout to make these changes.

- Remote checkin behavior.

  If you attach a local object to a replica using remote checkout, upon remote checkin, the attached local object can be transferred with ownership, can be sent as a replica, or cannot be sent at all. You can configure this behavior from the user interface and preference.

  From the **Remote Checkin** dialog box advanced options, you can select a relation to be transferred with ownership. The option to send a replica is not yet available in the user interface. Therefore, a relation listed in the **TC_remote_checkin_relation_send_as_replica** preference is sent as a replica.

  When you select a particular relation from the user interface, the following table explains the local attached behavior.

Relation included in user interface	Relation specified in preference	Behavior
Yes	Yes	Transfer of ownership
Yes	No	Transfer of ownership
No	Yes	Sent as replica
No	No	Not transferred

- Interoperability of certain business objects where subtypes have been added when it exchanges data with Teamcenter 8 may be problematic when the following occur:
  - A Teamcenter 8.3 site is upgraded from Teamcenter Engineering 2005 SR1 or Teamcenter 2007.
  - The upgraded Teamcenter 8.3 still exchanges data with Teamcenter Engineering 2005 SR1 or Teamcenter 2007 sites.
  - There are new subtypes created in the earlier Teamcenter Engineering 2005 SR1 and Teamcenter 2007 after the other sites in the federation are upgraded to Teamcenter 8.3.

The administrator should contact GTAC for assistance.

- Data exchange between Teamcenter 8.3 sites and Teamcenter 2007 MP1*x* or MP2*x* sites may generate errors in the following situations:
  - Execute **data_share -remote_import** from the Teamcenter 8.3 site for an object owned at a Teamcenter 2007 MP1*x* or MP2*x* site. The following error message is generated:

    ```
    Error 100403 - Operation 7 of application ITEM_APP is unsupported.
    ```

  - Check in a replica object from a Teamcenter 8.3 site for an owning Teamcenter 2007 MP1*x* or MP2*x* site. The following error is generated:

    ```
    Operation 12 of application RES_APP is unsupported.
    ```

  - Perform a remote import with transfer of ownership from Teamcenter 8.3 site for an object owned by a Teamcenter 2007 MP1*x* or MP2*x* site. The following error message is generated:

    ```
    POM INTERNAL ERROR - please report this.
    ```

  In these situations, the Teamcenter version does not support data exchange with Teamcenter 8.3. If possible, the administrator should upgrade the site to Teamcenter 2007.1 MP3 or a later version. Otherwise, the administrator should contact GTAC for assistance.

Part II *Configuring and administering Multi-Site Collaboration*

Chapter

4 Planning and setup

Chapter

4 Planning and setup

This information is intended for system administrators and other persons concerned with planning and setting up a Multi-Site Collaboration network. Other users need not be concerned with this information. Siemens PLM Software strongly recommends that anyone setting up a Multi-Site Collaboration network thoroughly review all planning considerations before performing any setup procedures.

With Multi-Site Collaboration you can set up a network of sites that can share data among one another on an as-needed basis. One of the primary concerns when using Multi-Site Collaboration is ensuring that the system is configured and maintained properly. This information is intended to provide some practical guidelines that you, as a system administrator, can use to plan your Multi-Site Collaboration network.

Advanced concepts

Getting started presents basic terms and concepts that all users, system administrators, and persons concerned with implementing a Multi-Site Collaboration network should understand. However, the following advanced concepts are essential for a thorough understanding of a Multi-Site Collaboration by technical audiences.

Integrated Distributed Services Manager (IDSM)

Multi-Site Collaboration solution discusses the concept of the Object Directory Services (ODS) and the role it plays in a Multi-Site Collaboration environment. Another fundamental component of Multi-Site Collaboration is the Integrated Distributed Services Manager (IDSM). While the ODS can be considered an object locator, the IDSM can be thought of as an object transporter. It provides the mechanism used to export an object from the owning site, transmit it over the network, and import it into the destination site. The IDSM functions the same when configured for remote procedure call (RPC) or HTTP/HTTPS communications except that there are no separate IDSM daemons as this functionality is part of the **tcserver** process. The calling site's logging functionality is the same for when using HTTP/HTTPS. However, logging that is done within the remote site's IDSM process for RPC communications is handled by the remote site's **tcserver** process when using HTTP communications. The content is the same but the log information is sent to the **tcserver** log file.

ODS and IDSM daemons

The ODS requires a server process or daemon. When using RPC communications, the IDSM also requires a daemon. The network nodes that run these daemons are referred to as the ODS or IDSM server node, respectively.

The ODS daemon is started by the **run_tc_ods** script and runs until the process is killed or the ODS server node is shut down. There is only one ODS daemon per ODS and it auto-logs in to the ODS database using the administrator user (**infodba** by default) 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 about two minutes, then terminates if it is not reused for another request.

You can have more than one IDSM daemon running on the same IDSM server node at a time. One IDSM daemon is required for each Multi-Site Collaboration request to deliver an object. This is an important factor to consider when configuring an IDSM server node.

Each IDSM daemon logs in automatically to the working site database that it serves using the administrator user account. For sites using rules-based object protection, Siemens PLM Software recommends 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.

Using remote procedure call (RPC)

When you configure Multi-Site Collaboration to use remote procedure call (RPC) technology for host-to-host communication, it is an important part of the setup process to ensure that the RPC software on your systems is operational outside of Multi-Site Collaboration.

The **rpcinfo** utility can be used at the operating system level to determine if the RPC software is operational. The following examples show how to use this utility:

- **rpcinfo -p** *node_name*

  This returns a list of RPC programs running at the specified node. For example:

Program	Version	Protocol	Port	Service
100000	4	TCP	111	Portmapper
100000	3	TCP	111	Portmapper
536875525	1	TCP	1035	Not applicable

  If no results or an error is returned, the RPC software was not installed correctly.

  If the Multi-Site Collaboration daemons are running, you should see some entries with the program numbers ending in **85** and **86**. Those that end in **85** are used by the ODS daemon and those that end in **86** are used by the IDSM daemon.

  For example:

Program	Version	Protocol	Port	Service
536875586	1	TCP	1035	Not applicable
536875585	1	UDP	1761	Not applicable

Program	Version	Protocol	Port	Service
536875585	1	TCP	2021	Not applicable

- **rpcinfo -T tcp** *node_name program_number version_number*

  Note

  On some platforms, the syntax of this command requires a lowercase **t** as in **rpcinfo -t**.

  Use this to test whether a daemon is ready, for example:

  ```
  program 536875586 version 1 ready and waiting
  ```

  If the daemon is not ready, the following message is returned:

  ```
  rpcinfo: RPC: Unable to receive; An event requires
     attention program 536875586 version 1 is not available
  ```

ISO/OSI network model

Multi-Site Collaboration integrates into the 7-layer ISO/OSI network model as follows:

- The Multi-Site Collaboration software resides in layer-7 (application layer).
- The RPC software resides in layer-5 (session layer).
- Multi-Site Collaboration also uses TCP and UDP protocols for layer-4 (transport layer).

Any networking enhancements below the transport layer (layer-4) are transparent to Multi-Site Collaboration. For example, you can use data compression and encryption enhancements with Multi-Site Collaboration without any changes to the software or the way it is installed.

Sharing write access to shared data

Data sharing does not involve modifying the shared data. Sites replicate a part for use as an assembly component with no intention of modifying the part itself. However, there are cases when a remote site must modify data owned by another site. In these situations, Multi-Site Collaboration provides two methods for sharing write access to shared data: *transferring ownership* and *remote checkin and checkout*.

Transferring site ownership

The remote site imports the object with transfer of site ownership. For items, this requires transferring site ownership of all revisions and most attachments and files. For item revisions with sequences, all sequences are transferred along with the sequence manager. Previous sequences are deleted from the transferring database. When the remote site gains ownership of an item, the item can be modified. When all modifications are made, site ownership is transferred to the original owning site or to any site that must modify the data.

Ownership access by remote users is controlled by the owning site using site preferences and access management rules.

If an item owned by Site1 is replicated to Site2, and the item's site ownership is transferred to Site3, the site ownership of the replica at Site2 is not updated to show the new owning site. Using the **data_sync** utility at Site3 does not update the replica at Site2, since the last modification date of the master copy at Site3 has not changed. It is not necessary to sync the owning site property because the replica at Site2 has not changed. To sync the replica at Site2, run the **data_share** utility at Site3 or perform a remote import at Site2.

Remote checkin and checkout

The remote site checks out the object by first replicating the item, then checking out the specific portion of the item requiring modification, for example an attached dataset. When the replica is checked out, a remote checkout is performed at the item's owning site ensuring no other user in the Multi-Site Collaboration network can modify it.

When all modifications are made to the replica, it is checked in to the owning site. All changes are sent to the owning site and the remote check out status is removed. Any new objects created are owned by the item's owning site.

This method avoids transferring site ownership of an entire item when write access is required only for portions of the item. For performance reasons, Siemens PLM Software recommends using this method whenever possible.

Note

For information regarding the types of data modifications available using remote checkin and checkout functionality, see *Modifying remote objects*.

Multi-Site Collaboration records

Multi-Site Collaboration uses replication to share data. This increases the need for keeping track of which sites have a copy of an object and when the copy was made. This information is stored in an Import Export Record (IXR). The IXR is a database object that is created during export and is attached to the master copy. When the master copy is modified, the information in the IXR is used to determine which copies must be synchronized.

The information in the IXR is also used to generate the **Exported To** property of a Teamcenter object. If you must see the information stored in an IXR, which includes the export reason, you can create a custom query on the **ImanExportRecord** class.

A similar concept applies when publishing an object to an ODS. When an object is published, a Publication Audit Record (PAR) is created and attached to the master copy. The information in the PAR is used to determine if an object needs to be republished, such as when the object's description is modified. If you must view the information stored in a PAR, you create a custom query on the PAR class.

Both the IXR and PAR objects reference the object they are attached to. This reference prevents the master copy from being deleted, ensuring network-wide referential integrity. To delete a master copy, the IXRs and PARs must be deleted first. PARs are deleted by unpublishing the object while IXRs are deleted using the **-verify** argument of the **data_sync** utility. For additional information, see the *Utilities Reference*.

Planning and setup checklist

The following table provides the preferred sequence of tasks for setting up a Multi-Site Collaboration network.

Task	Description
Review planning considerations	Review the *Planning considerations*. This helps you decide if Multi-Site Collaboration is the best data sharing solution for your enterprise and helps you plan your Multi-Site Collaboration network.
Fill out site information forms	Fill out one site information form for each site you intend to include in your entire (enterprise-wide) Multi-Site Collaboration network. For more information, see *Site information form*.
Configure Multi-Site Collaboration sites	Configure your working sites and at least one ODS site according to the instructions found in the installation manual for your platform.
Synchronize site definitions	Synchronize all site definitions by adding all site definitions in the entire Multi-Site Collaboration network to all Multi-Site Collaboration databases.
Synchronize POM transmit schema files	Distribute a copy of each site's POM transmit schema file to each site in the Multi-Site Collaboration network.

Planning considerations

The optimum Multi-Site Collaboration configuration varies greatly from enterprise to enterprise. This section helps you determine if Multi-Site Collaboration is the best data sharing solution for your enterprise, and the optimum settings for your enterprise-wide Multi-Site Collaboration network, should you decide to use Multi-Site Collaboration.

Determining when to use Multi-Site Collaboration

When implementing an enterprise-wide Teamcenter solution for your enterprise, it is easiest to use a single database for all users. However, when your enterprise comprises multiple facilities in different geographic regions, you must consider some sort of distributed Teamcenter solution.

It is possible to share data with various sites through the rich client's import and export functions. This solution is effective if you only share small amounts of data on a periodic basis. However, if you want to share large amounts of data on a regular basis you should consider using Multi-Site Collaboration.

Multi-Site Collaboration provides the publishing and system administration features needed to reliably share large amounts of data on a regular basis. Users can routinely search for and view data stored at other sites.

Site coupling

Teamcenter sites can be grouped into the following broad categories:

Site category	Description
Loosely coupled	Loosely coupled sites typically have little in common with one another on a day-to-day basis. For example, one site may perform design work and another site may perform manufacturing. Most of the work is completed by one site then passed on to another.
Moderately coupled	Moderately coupled sites are typically sites where multiple sites work together on a large product, but each site works on separate pieces of the product. For example, in an aircraft enterprise, one site may design the fuselage and another may design the wings.
Tightly coupled	Tightly coupled sites model a typical concurrent engineering environment where many teams at multiple sites work concurrently on the same part of the product.

Multi-Site Collaboration is the best solution for loosely and moderately coupled sites. It can support a tightly coupled site to some extent, but is not really intended to do so. In cases where you have tightly coupled sites, the best solution is to use a single database site that all teams access.

Planning your network

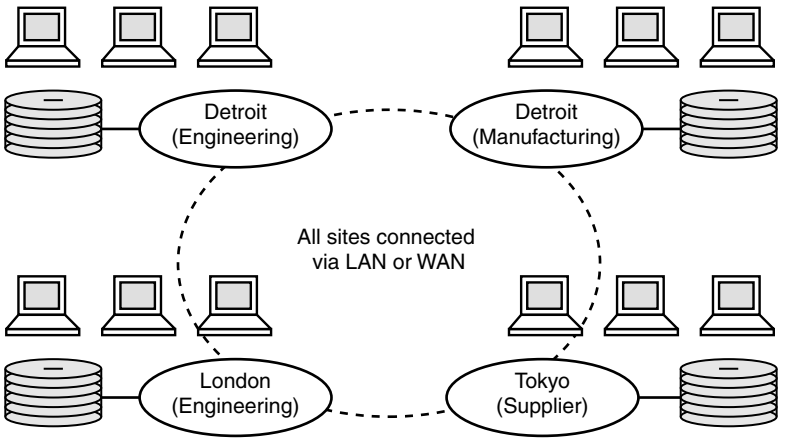
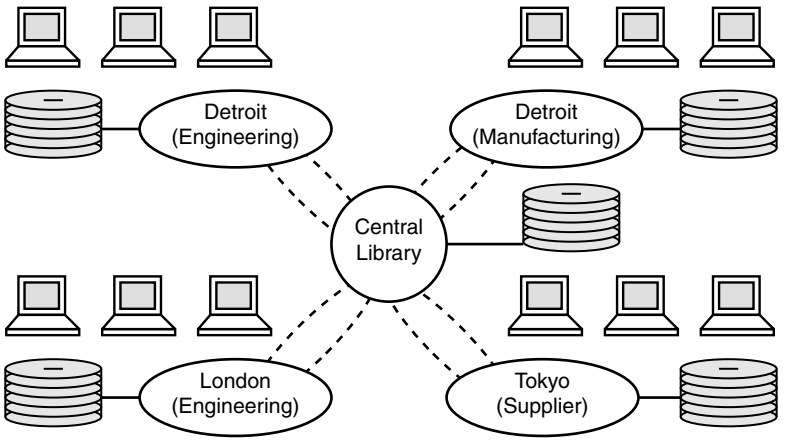
After you decide to use Multi-Site Collaboration, you must decide how many sites your enterprise-wide network includes and how to name them.

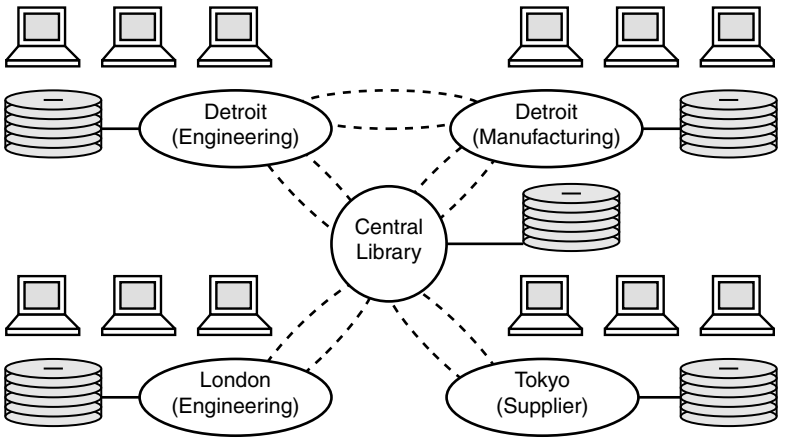
Multi-Site Collaboration network topology

The topology of a Multi-Site Collaboration network can be:

- Pure peer-to-peer
- Pure hierarchical
- Combination of peer-to-peer and hierarchical
- Hub configuration

For information on defining topology for internal company sites, see the recommendations in the following table.

Pure peer-to-peer	In a pure peer-to-peer topology, each site shares data directly with all other sites in the network. For this to occur, each site must be able to communicate directly and continuously with all other sites using a Local or Wide Area Network (LAN or WAN).  **Peer-to-peer**
Pure hierarchical	In a pure hierarchical topology, sites share data using one or more central libraries. Sites publish shared objects and transfer ownership of these objects to the central library. The central library contains all master objects used in the enterprise. Because all master objects are centrally located, sites do not need to communicate directly with each other. However, each working site must be connected to the central library through a LAN or WAN.  **Hierarchical**

Combination of peer-to-peer and hierarchical	A Multi-Site Collaboration network can also be configured as a combination topology using elements of both peer-to-peer and hierarchical topologies. You can decide which sites in your Multi-Site Collaboration network should be connected to one another to the central library through a LAN or WAN.  **Combination**
Recommendations	For internal company sites, start with a peer-to-peer topology so that the sites can freely share data amongst themselves. Then, to solve specific problems such as limiting supplier access, use a hierarchical topology by creating a supplier central library.

Global data caching

In some cases, you can improve performance by precaching structured context object (SCO) content at nondatabase sites and caching (instead of storing replica data in volumes) remotely imported data at a remote file server cache (FSC). If you have a large database at a particular site that contains some portion of the data that is accessed frequently by users at a remote site in a four-tier environment, you may want to prepopulate the replicas in the remote site's File Management System (FMS) cache. This is most useful when you have knowledge of what data a remote site works with frequently, such as when engineers at a certain site are responsible for certain assemblies or subassemblies of a product.

This capability can conserve volume space and improve Multi-Site utility performance by avoiding calls to the owning site during operating system (OS) level volume consolidation, such as when you consolidate smaller site databases into a single larger database.

You must consider the following caveats to using this approach:

- Precaching data is supported only within an enterprise and only on the FSCs defined in the local FMS master file (under *FSC_HOME*).
- Offline export from the first site to a second site with ownership transfer does not convert the **ImanFile** objects to **POM_stubs** objects.
- Operating system (OS) level volume consolidation (copy and move) is not supported across multiple enterprises.

The following Multi-Site utilities and commands can store replica files in a remote site's local FSC:

- **data_share** utility
- **data_sync** utility
- **Tools→Import→Remote** command
- **Multi-Site→Synchronization→...** commands

To enable the caching functionality for imports, you must set the **TC_force_remote_sites_exclude_files** site preference value to **true**. For exports, setting this preference to **true** causes the export directory to contain only the object metadata file. When the object metadata file is imported, Teamcenter creates augmented **POM_stub** objects with the attributes required to generate a read ticket enabling the importing site to generate read tickets from the stored information. However, the FSC at the importing site is not populated with the dataset files.

The augmented stubs must be synchronized when any of the following occurs:

- The owning site moves the dataset file from one location to another.
- The dataset file is transferred with ownership from one site to another.
- The dataset file is refiled.
- The dataset file is deleted or purged when a new version is created.

You use the **data_sync** utility with the **-sync_file_to_stubs** argument to synchronize **POM_stub** objects with the owning site **ImanFile** objects.

The **load_fsccache** utility generates read tickets and populates the target FSC. You can access this utility's functionality by using the **populatefsc** service that is accessed in the rich client through the **Translation→Translation** menu command. The **TC_validate_stub_tickets** preference, when set to true, allows Teamcenter to generate the latest stub ticket from the site where the file was transferred.

For more information about the using the **populatefsc** service, see the *Dispatcher Server Translators Reference Guide*.

You can conserve disk space in your Teamcenter database and volumes by using the **convert_replica_files_to_stubs** utility. This utility stubs replica **ImanFile** objects and purges the replica files from the corresponding volume.

ODS configuration

The Object Directory Services (ODS) site maintains a record of each object in the entire *Multi-Site Collaboration* network. The ODS does not store objects, it:

- Maintains a record for the object that is similar to a library card.
- Tells you which site is currently storing the object.
- Provides enough information about the object to allow you to decide if it is the object you want.

You can configure an ODS site to be either:

- Single process
- Multiprocess

  For a single server, ODS uses a single system process to service all incoming ODS requests from all sites. When most ODS-related requests are not database intensive, the single-process ODS server is the appropriate ODS configuration.

  For a multiprocess server, ODS creates subprocesses that perform the actual database operations, preventing the main ODS or a single subprocess from being fully occupied with a database intensive request. For example, this can occur when an ITK program is run to generate a report with several thousand records.

  Note

  Only one ODS license is used when running in multiprocess mode. The license is obtained by the parent process during the first incoming request.

If your ODS site is providing slow service due to a high number of ODS operations from remote sites, or if remote users are performing time-consuming operations, such as generating reports regarding published objects, consider running the ODS in multiprocess mode.

Configure a multiprocess ODS

By default, the ODS site is configured as a single server process.

1. Set the **ODS_multiprocess_mode** site preference to **true**. This preference is located in the **Data Sharing.Multi-Site Collaboration** category.

   For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

2. Set the **ODS_multiprocess_initial_subprocess_count** site preference to an integer. This preference defines the number of subprocesses the parent ODS in multiprocess mode created during startup. Define this preference only if you want to override the default value of **5**.

   For additional information about Multi-Site site preferences, see the *Preferences and Environment Variables Reference*.

3. Set the **ODS_multiprocess_max_subprocess_count** site preference to an integer. This preference defines the maximum number of subprocess that the parent ODS in multiprocess mode creates during startup. Define this preference only if you want to override the default value of **10**.

4. Stop the ODS server.

5. Restart the ODS server.

   The multiprocess ODS mode is implemented.

Hub configuration

A *hub* is a site with both an IDSM and ODS which acts as a clearing house between internal and external clients. A hub configuration is a method of integrating external sites, that is, suppliers and partners, and internal sites into a Multi-Site

Collaboration federation where the sharing of data is facilitated by the unique ability of the hub to replicate replicas in a controlled manner.

In a hub configuration, all data shared with external sites is replicated at the hub database and automatically published to its ODS. Suppliers need only search the hub ODS and can replicate a part directly from this central site, rather than from the actual owning internal site.

This configuration removes the requirement that external sites have direct network connections to internal sites, including the internal site ODS.

This configuration also improves overall network and system efficiency. By caching product data at a central location, the network traffic and the system load of the internal sites is greatly reduced.

A hub is beneficial in the following situations:

- Sharing data with second-tier suppliers.
- Sharing data between development partners.
- Creating a standard parts library.

You can simultaneously use a hub in one or all of the above situations.

You can set up multiple hubs. For example, as a site administrator, you can define multiple hubs within a Multi-Site Collaboration federation where one hub acts as a standard parts library, and another hub acts as a conduit to suppliers and development partners. You can also define a single hub that acts both as a library of standard parts and a conduit to multiple suppliers and partners.

Second-tier suppliers	Allows a supplier to securely provide replicas to subcontractors. **Scenario:** • As a supplier to Company A, Supplier1 accesses replicas directly from Company A. • Supplier1 subcontracts portions of a project to Supplier1-a and Supplier1-b. – Using a hub configuration, Supplier1 can designate its site to be a Multi-Site Collaboration hub and provide replicas to its subcontractors. – Without a hub configuration, Supplier1-a and Supplier1-b must retrieve replicas directly from Company A, which may not be acceptable to Company A.

Development partners	Allows all shared data to be replicated at a single location. **Scenario:** Two companies are development partners. Product components are produced by both companies; each company has multiple sites, all of which are involved in the development process. • Using a hub configuration, a single site can be designated as a hub. All shared data is replicated at the hub. Every site can access replicas directly from the hub. • Without a hub configuration, each site from one company would require direct network connection to each site at the other company to easily share data. Additionally, each site of one company would have to be individually defined in the database of each site at the other company.
Standard part library	Allows the creation of a central standard parts library. • Using a hub configuration, a standard parts library defined as a hub would be able to dispense replicas without having ownership of the parts. • Without a hub configuration, a standard parts library must own all the parts before it could dispense replicas of a standard part. The need to provide site ownership often prevents the creation of a library.

Finding hub data

You can find all shared data stored in a hub by searching the site ODS, because any object imported into the hub is automatically published to the hub ODS.

Hub ownership

A search for remote object on the hub ODS displays the remote objects as owned by the hub, not the actual owning sites. This guarantees that a subsequent remote import replicates from the hub, not the actual owning site which may be unknown to the importing site.

After importing from the hub, the replica objects show the hub as the owning site. This guarantees that reimporting the object using an import remote command, reimports from the hub, not the actual owning site.

Synchronizing replicas

To synchronize replicas, first run the **data_sync** utility at the owning site to synchronize the hub. Then run the same utility at the hub to synchronize the second generation replicas.

Working sites

Working sites are those sites in a Multi-Site Collaboration network other than ODS sites. They are found where normal users store their data. In order to participate in a Multi-Site Collaboration network, they must run IDSM processes.

This section contains the information for setting up working sites:

- General guidelines
- IDSM server node requirements
- Practical example of IDSM server node requirements
- Summary

General guidelines

When planning your working sites, use the following criteria to determine when to use separate sites (databases):

Criteria	Description
Geographical location	Siemens PLM Software recommends you use a single database for each facility in your enterprise (size permitting). This provides fast access to common data for all users at that facility. Avoid creating many small database (sites) at the same facility unless absolutely necessary. Consider using additional databases only to reduce server load at the same facility.
Size	If there are a large number of users at a single facility, it may be necessary to create several databases to reduce the load placed on a single database server. If this is the case, try to partition the users into functional work groups and assign entire groups to the same site (database).

IDSM server node requirements

The IDSM server node is the network node that runs the IDSM daemon for a particular working site. The IDSM daemon creates subprocesses to perform object copying from one site to another. These processes are short-lived and automatically terminate after a defined period of inactivity, usually two minutes.

To determine IDSM server node requirements, you must examine both the server hardware and the network link. This requires detailed analysis of the nature and size of shared data, frequency of data import and synchronization, and accounting for work patterns and schedules, including time zone differences, at all the sites.

Practical example of IDSM server node requirements

To provide a reliable general approach, consider the following two typical sites and how to compute the network traffic volume, megabytes-per-day, between these sites. Once this number is calculated, use it to estimate the megabytes-per-day between other sites. Ultimately, you can use all of these calculations to determine server hardware and network requirements at various sites.

The following procedure shows how to arrive at the network traffic volume for two Multi-Site Collaboration sites:

1. Identify common shared data types.

   Typically, this is the various items, such as nuts and bolts, that are shared between the two sites. It is best to deal with high-level compound objects such as items instead of individual objects such as datasets and forms.

2. Estimate the size of each shared data type in megabytes.

   For example, estimate the size of a bolt item and its revisions, using an average number of revisions, including the size of the metadata and any dataset file. Perform this process for each type of shared data. Also, maintain separate estimates for metadata and files. Later, you use this data to estimate database and volume disk requirements, respectively.

3. Estimate the number of high-level objects that travel from one site to the other per day and vice versa.

   There are three activities that contribute to this number:

   - Interactive remote imports from rich client.

     Estimate the number of times interactive users might pull a shared object over the network. You must calculate this for each shared data type.

   - Sending data using EPM handlers.

     Estimate the number of objects pushed by EPM handlers. You must calculate this for each shared data type.

   - Synchronization.

     Estimate the number of shared objects that must be synchronized per day based on the required synchronization frequency.

4. Estimate the megabytes-per-day in each direction.

   Based on the numbers obtained in the previous steps, estimate the megabytes-per-day in each direction. You should have separate numbers for the metadata and files and the total for both.

Summary of system network configuration

You now have numbers that you can use to configure your system and network. However, you must make adjustments based on work patterns and time zone differentials. You must ask questions such as *Does everyone pull parts from other sites when they come in at 8 a.m. and push released objects when they leave at 5 p.m.?*

Total megabytes-per-day implication	The total megabytes in both directions reveal bandwidth requirements for the network link between two sites. For example, if you determine that the total traffic volume in both directions is 500 megabytes-per-day, you need a high-speed link between these sites. If you cannot provide the necessary bandwidth (for example, for budgetary reasons), you must discuss reducing the network traffic by possibly excluding files or certain types of relations (such as manifestations) when transmitting objects. You can also use the total megabytes to help estimate the disk requirements for the IDSM server node. All network import/export operations involve using a local transfer area on the hard disk defined by the **TC_transfer_area** site preference. This disk should have enough capacity to handle a worst case scenario when many users are simultaneously transferring objects between sites. Allocating 10% of the total is normally adequate. However, remember that this is only for sharing data with one site. If an IDSM server node also communicates with other sites, then a similar amount of disk space must be allocated for each site.
Metadata megabytes-per-day and files megabytes-per-day implications	You can use the total size of the metadata copied to a site to estimate the incremental disk requirements for the database served by the IDSM. Similarly, you can use the total size of the incoming files to estimate the incremental disk requirements for the volumes. You must estimate how much of the incoming metadata and files is new (not due to synchronization). Again, this incremental data is for sharing data with one site. Add similar amounts of hard disk space for each additional site connected.
CPU and memory	The IDSM server node requires a heavy duty CPU with at least 128 megabytes of memory.
IDSM hosting recommendation	Because an IDSM server is always associated with a single database, some system administrators use the database node to host an IDSM server. This has advantages and disadvantages. The advantage is that import/export operations are faster because the database data is always on a local disk. However, it slows down non-Multi-Site Collaboration accesses to the database, and the networking activities could slow down the overall system performance. Therefore, unless you are willing to upgrade the database server node, Siemens PLM Software recommends not using it as the IDSM server node.

Object Directory Services (ODS) sites

Your Multi-Site Collaboration network requires at least one other site, an Object Directory Services (ODS) site. The ODS site provides librarian services for your Multi-Site Collaboration network.

There are several major planning considerations that you must consider and resolve:

- How many ODS sites should your enterprise use?
- What are the ODS server node requirements?
- What are the ODS networking requirements?

Number of ODS sites required

The optimum number of ODS sites for an enterprise is dictated by several factors which are described in the following table.

Item	Description
Number of sites	The number of working sites in an enterprise affects the number of ODS sites. If few working sites are involved, one ODS is sufficient. As the number of working sites increases, you must consider other factors.
Geography	If an enterprise has multiple working sites on several continents, it is best to maintain an ODS on each continent. This speeds up publishing and search operations. Even with these continental ODS sites, you can maintain a global ODS at a central strategic location in the enterprise.
Who needs to share data	As you add more working sites, it becomes increasingly important to carefully consider which sites share data among one another. If an enterprise has a wide variety of products, it is possible for a group of sites to share certain data among themselves while another group of sites share a different set of data. In this case, it is best to create and maintain separate ODS sites for each group.
Security consideration	Some enterprises are sensitive about letting suppliers have direct network connections to various sites and would prefer restricting access by suppliers to one specific site with a limited set of published parts. In this case, it may be necessary to create a separate *supplier* ODS containing only those publication records you want these suppliers to view.

ODS server node requirements

The ODS server node is the network node running the ODS daemon. Siemens PLM Software recommends that this node be separate from all other sites on the network.

The ODS basically responds to publication and remote search requests by accessing a single database table, the Publication Record table. No complicated queries are involved. Therefore, the ODS server node operations are not CPU, disk I/O, or memory-intensive.

The actual configuration of an ODS server node is dictated by three factors:

- Number of nodes that it serves
- Number of publication and remote searches
- Response time required

Item	Recommendation
CPU	Unless an ODS services hundreds of nodes and is constantly being queried, a reliable medium-sized CPU is sufficient.

Networking requirement

The network traffic between an ODS and the working sites that it serves generally consists of publication record data, for example, basic attributes such as ID, name, description, type, and class. In most cases, the network traffic is not heavy. Therefore, the link requirement between an ODS and a working site is fairly light. However, as with any distributed environment, extra network bandwidth usually improves performance.

Configuring a central library

A *central library* is normally used as an electronic vault to improve controls over released objects and to facilitate distribution of released objects to the various sites in the network. Typically, release procedures at the different working sites would transfer site ownership of released objects into the library. Sites that must replicate a shared object import it directly from the library.

It is possible to have more than one central library in a Multi-Site Collaboration network. For example, there can be a library for standard parts that is accessible to the entire company, another special parts library used only by one or two groups within the company, and another library for parts that suppliers must view.

Unlike an ODS which only stores publication records, a central library actually stores master objects used in the network. Therefore, it should be configured like a working site. However, there are some special considerations for central libraries.

Typically, users do not work directly on a library; a system administrator account, for example, **infodba**, is the only user account in the library. However, if the enterprise wants to preserve the identity of the original user and group that created an object, the library should define all user and group accounts that are sending objects to the library.

Because all sites are communicating directly with the central library and their primary purpose is to export or import data, the network link to a library should be a high-speed link. Disk drives should have fast access times and there should be enough memory to support multiple simultaneous requests from different sites.

Additional requirements for existing sites

The CPU, disk capacity, and memory requirements for each Multi-Site Collaboration site are primarily dictated by factors outside of Multi-Site Collaboration. Existing sites should already be properly configured for operation based on the number of users, expected volume of data, and third-party applications. However, it is still necessary to determine what incremental requirements Multi-Site Collaboration adds to the existing configuration.

For the node that hosts the database for a site, the largest impact is adding hard disk storage. As copies of objects from other sites are imported into your site, the database must grow to accommodate these objects. You must estimate the number of imported objects and their sizes and allocate additional hard disk space accordingly.

Siemens PLM Software recommends for general installation, you estimate the number of imported items and allocate 75 kB per item.

Additional memory and CPU upgrades are not necessary for the database host and the user workstations.

Site naming conventions

After you decide how many sites comprise your Multi-Site Collaboration network, you must decide how to name them.

Choose site names so that they are descriptive of the function or location of the site. For example, if the site is in Albany and all users working at that site share the same database, **Albany** is a suitable site name. If the site is known as the ABC Design Center, a name such as the **ABC Design Center** is better. The site name can contain up to 128 characters. Every site must have its own unique name and unique site ID. The site ID is defined automatically when the database is installed.

> **Warning**
>
> Do not change the site ID of a database once it is established. This site ID is used to generate internal identifiers for objects that must be unique throughout your enterprise. Also, do not reuse a site ID when creating a new database. For this reason, you cannot use database import and export tools to replicate a database; always use Teamcenter import and export.

ACS license considerations

It is important to determine the number of Multi-Site Collaboration Access Control Sheet (ACS) licenses required for the entire enterprise. If too few licenses are purchased, users cannot perform their work.

There are two types of Multi-Site Collaboration ACS licenses:

- ODS license
- Distributed User license

ODS license

The ODS license controls the number of ODS sites that can be run in the network. It is allocated when an ODS daemon starts and released when the daemon terminates. It is never checked during any Multi-Site Collaboration operation. You should purchase one ODS license for each ODS site you plan to run in the entire network. The total number can be placed in one ACS accessed by all ODS processes or it can be distributed to several ACS sheets.

Distributed User license

The Distributed User license is allocated whenever a user performs a Multi-Site Collaboration operation at a working site. The specific operations that require a Distributed User license are:

- Publishing and unpublishing an object
- Find remote

- Remote import
- Sending an object to another site using a release procedure

Each Teamcenter session requires only one Distributed User license no matter how many Multi-Site Collaboration operations are performed in that session. For example, if a user initiates a remote import and, while waiting for the import to complete, also publishes an object, only one Distributed User license is used as long as both operations are performed within the same Teamcenter session. Furthermore, ITK programs do not require a Multi-Site Collaboration license.

The Distributed User license is released immediately following each Multi-Site Collaboration operation. For example, if a user performs a **Find Remote** operation, the license is released immediately after the results of the search are obtained. If the user then decides to import one of the remote objects, a new license must be granted.

To determine the number of Distributed User licenses required, determine the number of users who are performing Multi-Site Collaboration operations simultaneously. It is not necessary to purchase a license for each user. However, purchasing too few licenses can hurt the users overall productivity (if all licenses are used up, users may have to wait for one to be released).

Working site security considerations

Security in a Multi-Site Collaboration environment is implemented in various ways depending on the level and nature of security wanted.

Site-level security is implemented using Multi-Site Collaboration preferences. These preferences allow you to define which sites can access data owned by your site. Other preferences allow you to take security a step further by defining which sites, if any, can transfer ownership of objects owned by your site. For further information about site-level security preferences, see the *Preferences and Environment Variables Reference*.

You can setup user-level security for a site by including the site in the **TC_check_remote_user_priv_from_sites** preference for the IDSM server.

For information about this preference, see the *Preferences and Environment Variables Reference*.

Access Manager (AM) validation is performed for user requests from sites that are set in this preference and the following site preferences are ignored:

- **IDSM_permitted_users_from_site**_*site-name*
- **IDSM_permitted_transfer_users_from_site**_*site-name*
- **IDSM_permitted_checkout_users_from_site**_*site-name*

For sites that are not included in this preference, the site-level security controls are used. Because this functionality is in the IDSM server, the user-level security control is applicable to all versions of Teamcenter and Teamcenter's engineering process management. However, the site preferences related to user-level security, AM rules, and the remote users must be added to all sites that are using this security mechanism. Also, the user data must be kept current.

User-level security is applied to the following Multi-Site Collaboration functions:

- Remote import
- Remote import with transfer of ownership

- Remote check out
- Remote export
- Remote export with transfer of ownership
- Data share utilities
- Pull synchronization
- On-demand synchronization

This security mechanism is recursive for BOM and distributed BOM operations. AM rules are applied to the child objects. However, when access to a child object is denied, the import behavior is not affected and is controlled by the **Continue On Error** option of the **Import/Export** dialog box. When a remote import request is received by a hub, the AM rules are applied at the hub.

User-level security allows AM rules to control access to an object by a remote administrator user for the following actions:

- Remote import
- Remote import with transfer of ownership
- Remote checkout
- Remote export
- Remote export with transfer of ownership

Local administrator user privileges (**infodba** by default) do not apply to a remote administrator user.

AM rules are validated for a remote user at the IDSM site for actions performed at a remote site as follows:

	Transfer out	**Export**	**Write**	**Import**	**Transfer in**
Remote import		Remote user		Remote site and optional **infodba** user	
Remote import with transfer of ownership	Remote user				Remote site and optional **infodba** user
Check out			Remote user and remote site		
Remote export				Remote user and remote site	
Remote export with transfer of ownership					Remote user and remote site

A remote administrator user can perform pull synchronization without import or transfer in privileges. For push operations, this permissions matrix applies to all

remote users that are defined in the local database of the IDSM site. No push operations are allowed for users that do not exist at the IDSM site.

You can implement object-level security using AM to protect individual objects that your site owns from unauthorized access by remote users. To control replication between working sites, you must set two accessors: **Site** and **Remote Site**. In addition, there are four export-related AM privileges you must be aware of: **EXPORT**, **IMPORT**, **TRANSFER_OUT**, and **TRANSFER_IN**.

The **Site** accessor refers to a specific site that you want to give or revoke a certain privilege. For example, you can grant a **TRANSFER_IN** privilege to the Detroit site so that users at that site can transfer ownership of objects from your site. The **Remote Site** accessor is the equivalent of the **World** accessor. World means *all users*, remote site means *all remote sites*. So if you grant **IMPORT** privilege to the **Remote Site** accessor, then you are granting the privilege to all remote sites defined in your database.

The **EXPORT** privilege (to export a read-only copy) and **TRANSFER_OUT** privilege (to export and obtain site ownership of master copy) apply only to the user that is performing the actual export of data. When you choose the **Command→Export→Objects** option to perform an export, the privileges apply to the user who is running the session. In the case of **Remote Import**, the privileges apply to the user account that is used to run the IDSM server, usually **infodba**.

The **IMPORT** privilege (to bring in a read-only copy) and **TRANSFER_OUT** privilege (to bring in a master copy) apply to the **Site** and **Remote Site** accessors only. These privileges do not apply to the user that is performing the operation.

There are several important things to remember about these accessors and privileges:

- All checks for the four export-related privileges are performed only at the exporting site. These privileges are not checked at the importing site. This gives the owner full control of the access privileges to the object because only local AM rules are used to control access. If the **IMPORT** and **TRANSFER_IN** privileges were checked at the importing site instead, the owner (who does not have control over privileges at remote sites) cannot control the access. Therefore, when users performing a remote import receive any export-related privilege error, the privileges at the owning site should be investigated.

- The **EXPORT** or **TRANSFER_OUT** (if transferring ownership) privilege is checked against the exporting user first, and if successful, the **IMPORT** or **TRANSFER_IN** privilege is checked against the importing site using the site accessors. If the second check is successful, only then is the object exported.

- The export-related privilege checks apply to each individual object that is exported and not only to the item. No export-related privilege checks are performed for non-Teamcenter objects. Checks for other privileges, such as **READ**, are performed as appropriate.

The desired object-level security that is appropriate for an enterprise is accomplished by defining appropriate AM rules for the accessors and privileges listed above.

So far, we have discussed ways to prevent users at remote sites from accessing your data. There are also ways to control what your local users can do as far as accessing remote sites:

- The remote sites that your local users can access are limited to the sites defined in your local database. Even if there are other sites that have physical connection to your site, your users can only access the sites that are defined in the local database.

- You can control the ability of individual users at your site to perform remote import operations from all sites or from specific sites. This can be easily accomplished this by defining appropriate AM rules.

- You can control which ODS sites your local users can search using the **ODS_searchable_sites** site preference that contains the list of ODS sites available to them. Even though users can save their own private list, the saved list must be a subset of the site preference list you have defined.

You can control which ODS sites your users can publish to. The list of authorized ODS sites consists of:

- The default ODS, as designated by the **ODS_site** site preference.

- The list of sites that are included in the **ODS_publication_sites** site preference.

While your users can search other ODS sites, as defined in the **ODS_searchable_sites** site preference, they can publish only to the authorized ODS list.

- You can disable, temporarily or permanently, publication from a site and still enable other Multi-Site Collaboration functions by setting the **TC_publishable_classes** site preference to **NONE**.

You can also control the ability to publish individual objects using the **PUBLISH** privilege that is indicated by the letter **P**. This privilege is incorporated in the default rule tree using the working named ACL which initially grants the **PUBLISH** privilege to the owning user only. The system administrator can extend this initial implementation to grant and revoke this privilege as desired.

ODS site security considerations

Information about a published object is stored in a *Publication Record* (PR) in the ODS. The publication record is never exported, so the export-related privileges cannot be used to protect the information.

As with working sites, you can implement site-level ODS security using site preferences that determine which remote sites can access an ODS. This type of security prevents a site from accessing the ODS for any purpose.

It is also possible to implement PR-level security where access to each publication record is controlled through AM rules. You must use the **Site** and **Remote Site** accessors to accomplish this. However, instead of dealing with export-related privileges, you grant or revoke **READ** privilege to a site accessor.

For example, you can define an AM rule such as *If owning site is Detroit, then only site Troy can access the Publication Record*. You do this by granting **READ** privilege to the **Site** accessor for Troy and revoke **READ** privilege from the **Remote Site** accessor.

For details on how to define AM rules to enforce ODS security, see *System administration*.

The PR-level ODS security, as implemented off-the-shelf, protects a publication record from all users from a remote site. If you want to implement a more granular security scheme such as protecting a publication record from specific users at a particular site, then you must implement the **USER_ods_check_pubrec_access** user exit.

Replica file management considerations

Files associated with replica objects require special consideration. Unlike metadata, files require considerable disk space. Because they are duplicates of the master copy, the need to store them on a long term basis becomes an issue. Replica files do not need to be backed up because a good copy can always be obtained from the master copy, if needed. In addition, after their initial use during the design process, the need to store these files at each replicating site becomes less and less important over time and at some point have to be deleted or compressed to save on disk space.

Multi-Site Collaboration addresses this problem by providing a means to segregate replica files into separate volumes. By default, all replica files are stored into the default volume of the importing user and are intermixed with non-replica files. You can segregate replica files by defining the **TC_replica_volume** preference to indicate the volume into which replica files are stored. This preference is defined as a site or group preference. It is possible to setup a separate replica volume for each group and at the same time specify a site-wide volume for those who do not have their own group replica volume.

For example, the Engineering group can define a group preference as:

TC_replica_volume=

eng_replica_volume

A site preference is defined as:

TC_replica_volume=

site_replica_volume

Once replica files are segregated, you use operating system tools to determine which files have not been accessed for some time and can be deleted or compressed. Once such replica files are identified, ITK programs can be developed to identify the objects associated with these files should it become necessary to delete replica objects that are no longer needed. When you delete a replica, you must first delete the dataset attachments before deleting the master object.

Transfer of site ownership security considerations

You can control the ability of remote users to transfer site ownership of objects owned by your site using site preferences and access rules.

Site preferences

The following two site preferences control the ability of remote users to transfer site ownership of objects owned by your site. For additional information regarding the use and behavior of these preferences, see the *Preferences and Environment Variables Reference*.

The system checks the following preferences before checking access rules.

Preference	Description
IDSM_permitted_transfer_sites	Defines which sites are authorized to transfer ownership of objects owned by the site served by an IDSM server. If not defined, no site is allowed to transfer ownership of any object from this site.
IDSM_permitted_transfer_users_ from_site_*sitename*	Defines which user IDs from the site specified by the above preference are authorized to transfer ownership of objects owned by the site served by an IDSM server.

Access rules

Grant or revoke the **TRANSFER_IN** privilege from the **Site** and **Remote Site** accessors to control the ability of remote users to transfer site ownership of objects owned by your site.

For example, if you want to grant Site B the ability to transfer site ownership, you would grant the Site B site accessor the **TRANSFER_IN** privilege. Alternatively, if you wanted to grant all remote sites the ability to transfer site ownership, you would grant the **TRANSFER_IN** privilege to the **Remote Site** accessor, instructing the system to grant the transfer privilege to all remote sites.

Remote checkout/checkin security considerations

You can control the ability of remote users to check out objects owned by your site using site preferences and access rules.

Site preferences

The following two site preferences control the ability of remote users to transfer site ownership. For additional information regarding the use and behavior of these preferences, see the *Preferences and Environment Variables Reference*.

The system checks the following preferences before checking access rules.

Preference	Description
IDSM_permitted_checkout_sites	Defines which remote sites are authorized to check out objects owned by the local site. If not defined, no site is allowed to check out any object from this site.

Preference	Description
IDSM_permitted_checkout_users_ from_site_sitename_	Defines which user IDs from the sites specified by the above preference are authorized to transfer ownership of objects owned by the local site. If this preference is not defined, all users from sites defined by the above preference may perform remote checkouts of objects owned by the local site.

Access rules

You can grant or revoke the **Write** privilege from the **Site** and **Remote Site** accessors to control the ability of remote users to check out objects from your site.

For example, if you want to grant site B the ability to check out objects from your site, you would grant the site B site accessor the **Write** privilege. Alternatively, if you wanted to grant all remote sites the ability to check out objects from your site, you would grant the **Write** privilege to the **Remote Site** accessor, instructing the system to grant the **Write** privilege to all remote sites.

Remote checkout privilege

The **Remote Checkout** privilege allows a user to check out objects that are not normally modifiable, such as a released item revision. The intended purpose is to allow additional attachments or other incremental changes that do not require write access to the object itself. If you import and check out an object that is not modifiable, the local object permissions show that you have write access even though it is unmodifiable at the owning site. Any changes to the local object cause the checkin to fail at the owning site.

The primary access check for a remote checkout operation is the **Write** privilege at the owning site. The **Remote Checkout** privilege can be used to allow the remote checkout of objects in which **Write** access is denied. Released objects are the most common example where this is useful. Remote checkout is allowed if the **Write** or **Remote Checkout** privilege is granted at the owning site. A side effect of this special behavior is remote checkout is permitted when the **Remote Checkout** privilege is denied if the **Write** privilege is granted.

An example usage scenario is you have released an item revision at the owning site and you want to be able to run an analysis or tessellation on the replica side, attach the output to the replica, and send the output back to the owning site. Because released objects are write-protected, you cannot remote checkout the revision to do this. The solution is to enable this operation by granting the **Remote Checkout** privilege to the revision at the owning site. Additionally, you need a way to get write access to the replica revision at the replica site, such as by using the bypass rule.

For information about access rules, see the *Security Administration Guide*.

Supporting multiple languages

The Teamcenter multilingual schema contains elements for localizable attribute value representations in one or more languages. This allows you to export and

import objects with localizable attributes for display names in more than one language. Teamcenter clients that access the imported data can display the localized attributes in differing languages depending on their locale. Prior to Teamcenter 8.2, Teamcenter sites were monolingual, that is, they could import attributes in only one language. Teamcenter 8.2 and later versions can be either multilingual or monolingual.

For information about displayed name localization, see the *Localization Guide*.

Consider the following when transferring objects with localized attributes and when monolingual and multilingual sites participate in your Multi-Site environment:

- Replicated objects cannot be modified at the replica site. Therefore, no translations can be added to an object's attributes except at the object's owning site.
- When transferring objects between multilingual sites, the data model representation for the localizable attributes must be the same at both the exporting and importing sites.
- For objects transferred between monolingual and multilingual sites, the localizable attributes must be exported in a language supported by the monolingual site.
- A **TC_master_locale**_*site-name* site preference must be set for each monolingual site at multilingual sites that export to the monolingual site. If this site preference is not set for a site, Multi-Site assumes the site is multilingual.
- A **TC_master_locale**_*site-name* site preference must be set for each monolingual site at all multilingual sites to allow the monolingual site to support publish and remote search operations.
- Standard Teamcenter multilingual sites have only the **pubr_objec_desc** attribute localized for **PublicationRecord** objects. You can customize the other publication record attributes, such as **pubr_objec_name** and **pubr_group_id** to allow publish and remote search operations on localized values for these attributes if required.
- Multi-Site transfers are supported between multilingual Teamcenter sites and Teamcenter sites which are version 2005 SR1 MP3 or later.
- Transferring objects with ownership from a multilingual site to a monolingual sites results in loss of localized values if the object is subsequently exported back to the multilingual site.
- The Teamcenter databases must be character set compatible. Therefore, transfers between sites using the UTF-8 character set and sites using non-UTF-8 characters are not supported.

Setup procedures

After you have reviewed the topics in *Planning considerations*, you can begin installing and configuring your enterprise-wide Multi-Site Collaboration network. Perform each of these procedures in the order that they appear in this section.

Configure Multi-Site Collaboration sites

Initial setup and configuration of Multi-Site Collaboration sites is accomplished using the installation script. Installing Teamcenter and configuring Multi-Site Collaboration is described in detail in the installation help for your platform.

The remainder of this topic briefly describes how to use the installation script to set up and configure Multi-Site Collaboration sites.

1. Install Teamcenter.

2. Create or upgrade a database for each working and ODS site.

3. Configure each of these databases (sites) for one of the following roles in the Multi-Site Collaboration network:

   These Multi-Site Collaboration configuration functions are found on the **Multi-Site Collaboration Configuration** main menu.

Choice	Description
IDSM	A normal database (site) running IDSM processes.
ODS	A special ODS database (site) running ODS processes.
Both	A combination database (site), extended to include ODS object classes, running both IDSM and ODS processes.

4. Perform postinstallation Multi-Site Collaboration configuration. This includes:

   - Multi-Site Collaboration environment preparation.

   - Configuring Multi-Site Collaboration daemons.

Configure FMS

Multi-Site Collaboration uses File Management System (FMS) during the file transfer process. You must configure FMS for each site in your network. Specifically, you must define a **multisiteimport** element to set the site IDs in the master file for each site. This file is located in the *TC_ROOT***fsc** directory for each Teamcenter site. The following figures show how to set these elements for two sites on the network:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fmsworld SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <multisiteimport siteid="459292456">
    <defaultfscimport fscid="FSC_cmh004_ntpriv_V710703" fscaddress="http:/cmh004:4544"/>
  </multisiteimport>
  <fmsenterprise id="1533032578">
    <fccdefaults>
      <property name="FCC_CacheLocation"
                     value="$HOME/V710703/FCCCache|/tmp/$USER/FCCCache"
                     overridable="true" />
      <property name="FCC_MaxWriteCacheSize" value="1000M" overridable="true" />
      <property name="FCC_MaxReadCacheSize" value="1000M" overridable="true" />
      <property name="FCC_LogFile" value="$HOME/fcc.log|/tmp/$USER/fcc.log"
                overridable="true"/>
      <property name="FCC_MaximumNumberOfFilePages" value="28672" overridable="true" />
      <property name="FCC_MaximumNumberOfSegments" value="10688" overridable="true" />
      <property name="FCC_HashBlockPages" value="6144" overridable="true" />
      <property name="FCC_MaxExtentFiles" value="11" overridable="true" />
      <property name="FCC_MaxExtentFileSizeMegabytes" value="256" overridable="true" />
    </fccdefaults>
    <fscgroup id="mygroup">
      <fsc id="FSC_cambr004_ntpriv_V710703"
           address="http://cambr004:4544" ismaster="true">
        <volume id="1b4c469ba12e5b603882" root="C:\\V710703\\gmssup_vols\\volume1" />
        <transientvolume id="68025247cf3591128889e2108807a0de"
                         root="C:\\V710703\\transientVolume_infodba" />
      </fsc>
      <clientmap subnet="127.0.0.1" mask="0.0.0.0">
        <assignedfsc fscid="FSC_cambr004_ntpriv_V710703" transport="lan" priority="0"/>
      </clientmap>
    </fscgroup>
  </fmsenterprise>

</fmsworld>
```

FMS master – site 1

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE fmsworld SYSTEM "fmsmasterconfig.dtd">

<fmsworld>
  <multisiteimport siteid="1533032578">
    <defaultfscimport fscid="FSC_cambr004_ntpriv_V710703"
                     fscaddress="http://millabv22:4544"/>
  </multisiteimport>
  <fmsenterprise id="459292456">
    <fccdefaults>
      <property name="FCC_CacheLocation"
                value="$HOME/V710703/FCCCache|/tmp/$USER/FCCCache"
                overridable="true" />
      <property name="FCC_MaxWriteCacheSize" value="1000M" overridable="true" />
      <property name="FCC_MaxReadCacheSize" value="1000M" overridable="true" />
      <property name="FCC_LogFile" value="$HOME/fcc.log|/tmp/$USER/fcc.log"
                overridable="true"/>
      <property name="FCC_MaximumNumberOfFilePages" value="28672" overridable="true" />
      <property name="FCC_MaximumNumberOfSegments" value="10688" overridable="true" />
      <property name="FCC_HashBlockPages" value="6144" overridable="true" />
      <property name="FCC_MaxExtentFiles" value="11" overridable="true" />
      <property name="FCC_MaxExtentFileSizeMegabytes" value="256" overridable="true" />
    </fccdefaults>
    <fscgroup id="mygroup">
      <fsc id="FSC_cmh004_ntpriv_V710703" address="http://cmh004:4544" ismaster="true">
        <volume id="1b4c469ba7ed1b603f28" root="F:\\V710703\\gmsnsn_vols\\volume1" />
        <transientvolume id="79f4d0f3de7571e9d4db7452bece1f79"
                         root="F:\\V710703\\transientVolume_infodba"/>
      </fsc>
      <clientmap subnet="127.0.0.1" mask="0.0.0.0">
        <assignedfsc fscid="FSC_cmh004_ntpriv_V710703" transport="lan" priority="0" />
      </clientmap>
    </fscgroup>
  </fmsenterprise>

</fmsworld>
```

FMS master – site 2

For information about FMS, see the *System Administration Guide*.

Configure global data caching

You must specify the **exitsfsc** elements (target file server caches) where you want replica files cached. You do this using the **populatetargets** attribute of the element. This attribute is a string of arbitrary names that is used to identify **exitfsc** objects for population. You execute the populate command with a list of one or more populate targets that reference some number of **exitfsc** objects.

1. In My Teamcenter, set the **TC_force_remote_sites_exclude_files** preference to **true**.

   For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

2. Open the file server cache (FSC) configuration file and add a value for the **populatetargets** attribute to the **exitfsc** element, for example:

   ```
   <exitfsc fscid="fsc1" populatetargetids="default,all"/>
   ```

For more information about configuring FMS for global data caching, see the *System Administration Guide*.

Set up a hub

A hub is a site with both an IDSM and ODS that acts as a clearing house between internal and external clients. A hub configuration allows the replication of replicas in a controlled manner.

In a hub configuration, all data shared with external sites is replicated at the hub ODS. Suppliers need only search the hub ODS, and can replicate a part directly from this central site, rather than from the actual owning internal site.

This configuration removes the requirement that external sites have direct network connections to internal sites, including the internal site ODS.

This configuration also improves overall network and system efficiency. By caching product data at a central location, the network traffic and the system load of the internal sites is greatly reduced.

For additional information about the benefits and requirements of a hub configuration, see *Hub configuration*.

1. During base installation, set up the site as both an IDSM and ODS site.

2. Define a site as a hub site:

   a. From the Organization application, select **Sites** from the **Organization List** tree.

   b. Type the values in the **Site Name** and **Site ID** boxes as you would normally for a regular site.

      Note

      Siemens PLM Software recommends that the site name includes a prefix or postfix of hub. For example **mycompany-hub**. The site name value can contain up to 128 characters.

   c. Select **Provide Object Directory Services**.

d. Select **Is A Hub**.

e. (Optional) If you are using HTTP or HTTPS instead of RPC for communication, select **Is HTTP Enabled**.

f. If you select **Is HTTP Enabled**, type the URL of the Teamcenter Web application in the **Node Name** box.

g. Click **Create**.

3. Define and assign values to the **IDSM_permitted_sites** site preference. Each client site must be defined in this preference.

   For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

4. Define and assign values to the **IDSM_permitted_transfer_sites** site preference for any site to be allowed to transfer site ownership from the hub.

5. Modify the **ODS_site** preference; assign it the site name of the hub site.

6. Define the **ODS_searchable_sites** site preference to include all ODS sites accessible to the hub, including the hub ODS.

7. Modify the site preferences for other Multi-Site Collaboration preferences, such as the **TC_transfer_area** site preference.

8. Enable the hub to retain group ownership of replicas by defining the pertinent group names, then define the groups in the **TC_retain_group_on_import** site preference and set it to **TRUE**.

   Caution

   If the group set in this preference is not defined at the importing site, this preference has no effect and the group is set to the default group of the user doing the import.

9. Populate the hub using the **data_share** utility, or by performing a remote import operation from the hub.

To set up a hub client in the rich client:

1. Install the base software and configure the client site as a regular IDSM site.

2. Define the remote hub site as a hub and as an ODS service provider:

   - If the hub site is not defined in the local database, enter:

     ```
     $TC_BIN/site_util -f=create -site=id=123456789
        -site_name=detroit_hub -node=sun135 -ods=y -hub=y
     ```

   - If the hub site is defined, enter:

     ```
     $TC_BIN/site_util -f=modify -site=id=123456789
        -ods=y -hub=y
     ```

3. For each client site, edit the **IDSM_permitted_sites** preference to include the hub site.

For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

4. For each client site, edit the **ODS_searchable_sites** site preference to include the hub site. The hub client is configured.

The hub client is configured.

Synchronizing site definitions

After all sites in the Multi-Site Collaboration network are created and configured, each site must add site definitions to its database for all other sites.

For example, Detroit-Manufacturing automatically has its site definition added to its database during installation. Following installation, the Detroit-Manufacturing system administrator must manually add site definitions for the rest of the Multi-Site Collaboration network (for example, Detroit-Engineering, London-Engineering, Tokyo-Supplier and SaoPaulo-Supplier).

Warning

All site definitions must be added to all databases in the Multi-Site Collaboration network and they must be identical. This requires close coordination among all system administrators in the entire Multi-Site Collaboration network.

Synchronizing POM transmit schema files

The final Multi-Site Collaboration setup procedure involves synchronizing Persistent Object Model (POM) transmit schema files among all sites in the entire Multi-Site Collaboration network. Basically, this involves physically copying each site POM transmit schema file and sending it to every other site in the Multi-Site Collaboration network. These copies are stored in the **$POM_TRANSMIT_DIR** directory.

Caution

When IDSM or ODS is configured to run as a Windows service, you must use a UNC formatted path for the **POM_TRANSMIT_DIR** variable. If you use a network drive (mapped) letter in this variable, the service is not able to locate the directory to read the required files.

Warning

Certain sites may be using more than one POM transmit schema file. This is because 64-bit and 32-bit platforms require separate POM transmit schema files. Copy all applicable current POM transmit schema files.

For additional information about POM transmit directory variable and schema files, see the *Preferences and Environment Variables Reference*.

Setting up partial item export

You can use various options to control the parts of an item that are exported and subsequently imported. In most cases, the user at a remote site performing the remote import operation can choose the parts of an item to import/export

(for example, when the latest revision of an item is requested). However, some import/export options should be controlled at the owning site using the Access Manager. When planning your Multi-Site Collaboration network, you must take this information into consideration.

The owning site must consider which options they want to use to control the components of an assembly that can be exported:

- Selected revisions only
- To exclude export protected components of an assembly

Both of these options enable the owning site to control which parts of an item or assembly that remote sites can replicate. For example, assume an item has three revisions A, B, and C:

- Revision A can only be replicated by Site1
- Revision B can only be replicated by Site 2
- Revision C is open to all remote sites

Using the Access Manager, you revoke the **IMPORT** privilege from Site 1 for revision A and revoke the **IMPORT** privilege from Site 2 for revision B. When users at Site 1 and Site 2 try to replicate the item, they receive only the revisions that their site is allowed to import.

The option to exclude protected components in an assembly works in the same manner described above, except the protection applies to the component items instead of individual revisions. Use this option to avoid the constant reimporting of component items that are stable and widely-used. For example, particular items have been replicated to all sites and are not likely to change in the future. Without export/import protection, such a component can be unnecessarily exported and imported when an assembly or subassembly containing it is synchronized.

When planning your Multi-Site Collaboration network, you must identify these stable and widely-used components and deal with them accordingly.

Warning

Although Access Manager provides various ways to protect objects from import/export using accessors, Siemens PLM Software recommends that you use only the site accessors. This makes the operation of these options insensitive to the user context of the IDSM process.

Setting up data synchronization

After an object or item is replicated to other sites and then subsequently modified, it is necessary to update the replicas. The process of updating replicas is referred to as *synchronization*.

There are several import/export and synchronization options to help you carefully plan and set up synchronization.

Planning data synchronization

There are important factors that you must consider when planning data synchronization.

Pull versus push strategy

Synchronization is accomplished through either a *pull* or a *push* strategy.

- The pull strategy uses the **Import Remote** command to update a replica. This command refreshes the replica by reimporting the information from the owning site.

- The push strategy uses the **data_sync** utility and the **Automatic Synchronization** facility to determine the changes that were made in the master copy, and then pushes the modified objects to update the replica.

With the introduction of partial item export, it is important to understand the pull and push strategies. When importing the latest revision of an item from a remote site, it is important to synchronize the item so only the latest revision is updated, instead of updating the whole item by bringing in all the other revisions. In addition, it is important to synchronize only those attachments that were requested by the importing site.

- Using the pull strategy, you can ensure that only the replicated revision attachments are updated by specifying the same options you used during the initial replication operation.

- Using the push strategy, Multi-Site Collaboration uses item export record information as the basis for synchronization. The item export record is created for each exported item to record the import/export options used the last time the item was exported to a given site. The stored options include the revision selector, including the release status type if this revision selector was used, the list of excluded attachment types, and dataset version and file options. This guarantees that the same options and attachments are used during synchronization.

The **data_sync** utility and the **Automatic Synchronization** facility both employ the push strategy. In both cases, the default synchronization technique is to use the information in the item export record as the basis for determining objects to be synchronized. For the **Automatic Synchronization** facility, there is means of overriding these defaults. However, for the **data_sync** utility, it is possible to override these defaults.

Using revision selectors

By default, the **data_sync** utility synchronizes only the latest revision of an item (**latest_revision**) when synchronizing to multiple sites. If you are synchronizing to a single site, the selector used the last time the item was exported (**same_as_last_export**) is used. If these modes are not appropriate for your installation, you can override them by specifying one of the revision selectors available with the **data_sync** utility. The following mutually exclusive revision selectors work with the **data_sync** utility:

Revision selectors	Description
all_revisions	Synchronizes all revisions of an item.
all_released_revs	Synchronizes all revisions with a release status including in-process item revisions.
latest_working	Synchronizes only the latest working revision of an item, if any.
latest_working_or_any	Synchronizes only the latest working revision of an item, if any. If an item has no working revision, the latest released revision is synchronized.
latest_released	Synchronizes only the latest released revision of an item. Use this in situations when sending the latest released revision is important (for example, when updating parts previously sent to suppliers).
release_status=*rstatus*	Synchronizes only the latest released revision of an item with the given release status. Use this in situations when sending the latest released revision with a specific release status is important (for example, when updating parts with a specific status previously sent to suppliers).

Warning

These revision selectors and the default latest revision switch apply only when the **data_sync** utility is synchronizing objects of the **Item** class. This occurs as a result of either the **-class=Item** switch or the other switches that specify items for synchronization.

Synchronizing specific revisions

When synchronizing objects in the **Item Revision** class, use either the **-class=ItemRevision** switch or the **-filename** switch of the **data_sync** utility to synchronize the specific item revision and all its attachments.

The **-class=ItemRevision** switch uses the modified-since-last-export rule to determine the synchronized revisions. This means that an object is selected based on whether it was modified after the last time it was exported to the sites involved. If so, the revision synchronizes regardless of the revision selector specified. The parent item is also synchronized.

Synchronizing a single site versus multiple sites

You can perform synchronization one site at a time or for multiple sites. Typically, it is more efficient to synchronize multiple sites rather than a single site. An item to be synchronized is exported once and then sent to all the sites, instead of exported to each site and then sent to each destination. When you run the **data_sync** utility at Site1 to synchronize both Site 2 and Site 3, certain complications can occur:

- If **data_sync** is synchronizing the item (which by default sends the latest revision only), then revB is exported and sent to Site 2 and Site 3. Site 3 receives the revision it requested, but Site 2 does not. Site 2 received revB (which was not requested) and does not receive the requested revA updates.

- If **data_sync** is synchronizing revisions and both revA and revB were modified, then both revA and revB are exported and sent to Site 2 and Site 3. Now, Site 2 and Site 3 received revisions they did not request.

An enterprise should choose a strategy that is appropriate for the way it does business. The multiple site synchronization strategy is suitable for sites that have a common replication needs.

The single site synchronization strategy is appropriate for those where every site can have different replication needs.

The recommended approach is to group sites that have similar synchronization requirements and do a multiple site synchronization for these sites. Use a separate run of **data_sync** to do single site synchronization for sites that have unique synchronization requirements. For example, Site 1, Site 2, and Site 3 all have the same synchronization requirements, and Site 4 has a unique requirement. Siemens PLM Software recommends you run **data_sync** twice:

- **$TC_BIN/data_sync -site=Site1 -site=Site2 -site=Site3 -class=Item ...**

- **$TC_BIN/data_sync -site=Site4 -class=Item ...**

Synchronizing a single class and multiple classes

When running the **data_sync** utility, you can enter one or more classes at the command line. When using a single class, you must run the **data_sync** utility multiple times until all classes of data are synchronized.

Whether it is better to use a single class and or to use multiple classes varies with each installation. It is important to perform tests when you first set up Multi-Site Collaboration (and later on if you feel that synchronization is taking too much time) to see which scheme is appropriate for your installation. Use the following as a guide:

- If the number of objects to be synchronized are relatively small, for example, in the low hundreds, then using multiple classes in a single invocation of the **data_sync** utility is recommended.

- If the number of objects to be synchronized is relatively high, using a single class with multiple invocation is recommended. This prevents the **data_sync** utility from loading too many objects in memory which slows down the synchronization operation.

A related issue is the order in which the different classes should be synchronized. As a rule, you start synchronizing the high-level objects (items) moving to low-level objects (forms and datasets). The rationale is that synchronizing high-level objects would automatically synchronize low-level objects that are attached. Thus, synchronizing a dataset first would likely resynchronize it again when the item is synchronized. When the item is synchronized first, any attached dataset does not need to be synchronized when the dataset class is processed.

Synchronizing by class or file name

The **-filename** switch enables the synchronization of specific items whose item IDs are contained in a specific file. The list of objects to be synchronized are placed in an operating system text file. The **-filename** switch reads the name of the system file and gives it to the **data_sync** utility. The **-classoffile** switch provides the **data_sync** utility with the class of the object in the list.

While the **-class** switch selects objects based on the modified-since-last-export rule, the **-filename** switch makes it possible to synchronize a specific set of objects. The **-item_id** switch is used to support wildcard entries and to further facilitate the synchronization of items.

The **data_sync** options, **-filename** and**-item_id**, make it possible for a site to implement a more efficient means of synchronization, depending on the needs of cooperating remote sites. For example, if it is known that a given remote site shares only items with IDs that start with a certain prefix, then using the **-item_id** switch can improve performance substantially.

Synchronizing assemblies or individual items

By default, the **data_sync** utility synchronizes only individual items and does not traverse down an assembly tree. Synchronizing an entire assembly is very inefficient especially when all revisions of every component are synchronized.

With all of the enhancements related to exporting of items, synchronization of entire assemblies is an efficient alternative in certain situations. For example, if two sites are sharing data related to a particular assembly or a limited number of known assemblies, then it is more efficient to synchronize by assembly, rather than by individual items.

Use the **-include_bom** switch to synchronize by assembly. You give specific assembly or assemblies through the **-filename** switch or the **-item_id=** switch.

The **-include_bom** switch is also relevant when synchronizing **BOM viewRevisions** (BVR) through the **-class=PSBOM viewRevision** switch.

- Without the **-include_bom** switch, the **data_sync** utility synchronizes only the specific BVR.
- With the **-include_bom** switch, the **data_sync** utility traverses to the component items and sends the latest revision, or the revision selector specified, and adds any new components found to the assembly.

Synchronizing modified objects only

There are situations when this strategy may not result in optimum efficiency. You must understand that the modified-only option involves some preprocessing. The system determines if an object was modified since the last time it was exported to a particular site. It processes the export record (IXR) associated with each exported object. There is one IXR for every site that the object was exported.

Generally, if the preprocessing time is significantly less than the time it takes to blindly export an object (and related subobjects), the modified-only option results in a more efficient operation. However, as the number of sites being synchronized (multiple-sites) increases, the number of IXRs to be checked increases along with the preprocessing time.

At some point, the preprocessing time becomes significant enough that it is more efficient to blindly export an object and its attachments. This varies from site to site depending on the size of dataset files and the number of attachments. When this occurs, use the **-disable_modified_only** switch to override the modified-only option.

As a rule, system administrators should refrain from using the **-disable_modified_only** switch, but should use it when analyzing efficiency problems associated with the **data_sync** utility.

Setting up data compression

A **Remote Import** operation, a release procedure that sends a released object to a remote site and synchronization using the **data_sync** utility all have one thing in common, data is exported from the owning site, data is transmitted through the network, and the receiving site imports the data.

The export and import data operation is disk-input and output intensive and generally executes fast. Transmitting data over the network creates a bottleneck under certain conditions. Typically, these conditions occur when two remote sites that are connected to a wide area network (WAN) must transmit large chunks of data such as datasets with NX part files. The Multi-Site Collaboration data compression feature improves the transmission efficiency between such sites.

By default, data compression between two sites is disabled. To enable the data compression feature at a site, the following two definitions are required:

- Set the site preference **IDSM_Compression** value to **TRUE**.
- Set the site preference **IDSM_Compression_Type** to **InfoZip**.

For example:

```
IDSM_Compression = TRUE
IDSM_Compression_Type = InfoZip
```

For compression to take effect between two working sites, both sites must enable data compression and the compression types must be identical. If one site has it enabled, but the other site has it disabled, then compression is disabled between the sites.

This feature makes it possible to employ data compression between two sites only when needed. For example, a Multi-Site Collaboration network has three sites: Site 1, Site 2, and Site 3. Site 1 and Site 2 are connected by a WAN and Site 3 is connected to Site 1 through a local area network (LAN). Site 1 and Site 2 both enable data compression to optimize the WAN transmission. Site 3 disables compression because it is connected to Site 1 through a LAN. Although Site 1 has enabled compression, it employs the compression only when exchanging data with Site 2, but not when exchanging data with Site 3.

When synchronizing data with multiple sites, follow this important rule: Synchronization between sites with data compression enabled must be performed separately from sites that have data compression disabled, such as separate runs of the **data_sync** utility. In the example above, since compression is enabled between Site 1 and Site 2, but is disabled between Site 1 and Site 3, the **data_sync** utility must be run at Site 1 separately for Site 2, and Site 3.

On-demand synchronization

On-demand synchronization, either through the rich client or the **sync_on_demand** utility, uses the selected revision rule that is an existing revision rule defined at the local site. Its name is passed to the owning site of the selected component and is used by the owning site to determine the item revision to synchronize. It is required, if the selected object is an item. By default, the list of revision rules is the set of all revision rules defined in the local database. However, this can be overridden by the **TC_sync_revision_rules** site preference. In this preference, enter a list of revision rules that appears in the rich client as the **Specific Revision Rule** list on the **Synchronization preferences** dialog box. If this preference is not defined, the default is to use all the revision rules defined in the local database in the list.

The synchronization on-demand report function tracks all unavailable sites (any site that times out) identified during the report activity and does not send additional queries to those sites during the session. Two site preferences control the behavior of the report function of on-demand synchronization.

- **Report broadcast mode**

  The **TC_on_demand_sync_broadcast_mode** preference controls the query scope of the report function when the site known by the current site as the owning site denies ownership. If it is set to **TRUE** (default value), the function queries all known sites to find the owner (broadcast mode). When set to **FALSE**, the function performs sequential queries to sites in the ownership chain until it reaches a designated limit (**TC_follow_ownership_chain_max_site_count** preference).

- **Ownership chain limit**

  The **TC_follow_ownership_chain_max_site_count** preference sets a limit on the number of sites that are queried before a replica's owner is designated as **unknown**. When the site known as the owning site returns a new owning site, the report function queries the new owning site for the replica's state. This activity continues until the owning site returns the replica's state or the value set in the preference is reached. When the limit is reached the function checks the **TC_on_demand_sync_broadcast_mode** preference to determine whether to use broadcast mode or to designate the ownership as **unknown**. The default value is **No limit**.

ODS security

When an object is published, a publication record is created at the ODS site. This publication record contains most of the relevant information about the published object, such as ID, description, and the owning site. When a user uses **Find Remote** to search for published objects, **Find Remote** scans the publication records to find the records that match the search criteria. For some enterprises, publication records represent sensitive information that are secured very much like the published objects themselves. For example, a company may require that publication records for parts manufactured internally are not accessible to external suppliers. In some cases, a particular external supplier is not allowed to even know about parts contracted to other suppliers.

Multi-Site Collaboration's ODS security mechanisms use AM rules on publication record attributes. For example, you can define a rule, such as *If the owning site in the publication record is Detroit, then only the Troy and Ohio sites have READ access*

to the publication records. When the owning sites receive a query with the correct search criteria, but the site is not authorized, the end user cannot access the record.

Warning

When defining the AM rules, the **READ** privilege for a publication record is the only relevant privilege.

When planning your Multi-Site Collaboration network, you must gather information about the ODS security requirements of your different sites. In some cases, security requirements are dictated by contract terms with partners or suppliers. The standard Multi-Site Collaboration software allows you secure publication record access at the site level, such as prevent **READ** access for all users at a particular site. If you must control access at a lower granularity, such as prevent access to Joe at the Michigan site, then use the **USER_ods_check_pubrec_access** user exit to implement this rule.

Configure remote inboxes

Remote inbox functionality is dependent on Teamcenter interoperability. To enable the Teamcenter-to-Teamcenter remote inbox feature, you must set the **TC_external_app_reg_url** preference to the URL where the **ApplicationRegistry.war** file is deployed. The URL must be in the following format:

http://*appserver-host***:***port-number***/ApplicationRegistry**

If this preference value is not set, you get an error when you attempt to subscribe to a remote inbox that states the option value cannot be found.

For information about installing and deploying the Application Registry application, see the *Teamcenter Interoperability* guide.

Teamcenter administrators with valid WebKey accounts can access the *Teamcenter Interoperability* guide at the following location:

http://support.ugs.com/docs/tc_eng/8/en/tss00004.pdf

For information about using remote inboxes, see the *Rich Client Interface Guide*.

Chapter

5 Synchronization

Chapter

5 Synchronization

A replication-based solution must ensure that replicas are kept up to date when the master object is modified. Multi-Site Collaboration addresses this by maintaining export records and providing synchronization tools. The process of keeping replicated data up-to-date is called synchronization. You can synchronize data between sites manually or automatically.

Export records

When an object is exported, export records are created for each target site specified. Each export record contains the site ID of each target site and the date of the last export to that site. Export records are always associated (and stored) with the master object. For items, a special Item Export record is also created to record the Import/Export options used so that these same options can be used to synchronize the Item.

data_sync utility

When the master object is modified, replicas can be updated by an administrator through the **data_sync** utility.

For information about using utilities, see the *Utilities Reference*.

Synchronization

When a master object is replicated at other sites, it is necessary to update the replicas whenever the master object is modified. There are important factors to consider when planning data synchronization. Site administrators should review *Planning data synchronization* for planning information regarding data synchronization to determine which types of synchronization should be used.

Data synchronization options

Synchronization options are set in the **Import Remote Options** dialog box. You can choose between *automatic* synchronization and *batch* synchronization. You can also choose to be notified when the master object is modified.

Choose automatic synchronization when you have imported a replicated object and want to specify that your replica is to be synchronized immediately after the master object is modified. This results in an efficient and evenly distributed synchronization process in which replicas are updated minutes after the master copy is modified.

Additionally, request to be notified when the master object of your replica is modified by selecting the **Notify by E-mail** option. When you subscribe to the **Replica Updated** event for a replica object at replica site, you are notified when the replica gets updated due to a reimport or synchronization. However, the e-mail notification is sent only for the objects of classes listed in the **TC_subscribable_replica_classes** preference. Therefore, this preference must include the names of all the classes of objects for which you require update notification e-mails.

Note

Automatic synchronization can only be used when importing remote objects; it cannot be used when performing interactive object export.

Choose batch synchronization when you import a replicated object and want the administrator at the owning site to synchronize your replica with the master object. The synchronization is performed using the **data_sync** utility; your replica and any other replicas defined for the utility are synchronized in a single batch. When you choose this method, the synchronization is performed at a time scheduled by the owning site administrator.

Option	Results
Synchronize Automatically	Imported replicas are synchronized automatically when the master object is modified. This option can be used at any site. If the **Notify By E-mail** option is also defined, you are notified when the master object is modified. For information on enabling this functionality, see *Enabling automatic synchronization*.
Synchronize in Batch Mode	Imported replicas are synchronized only when the **data_sync** utility is run. This utility can only be run by the owning site. If the **Notify By E-mail** option is also defined, you are notified when the master object is modified. For information about using this utility, see the *Utilities Reference*.
Notify By E-mail	You are notified by system e-mail when the master object is modified. This option creates a subscription at the owning site and on the replica. The subscription contains a notification handler that performs the actual notification at the replica site. Subscriptions are not created for objects within an item. System e-mail is enabled using the **Mail_server_name** site preference and **TC_subscribable_replica_classes** preference. For information about defining these preferences, see the *Preferences and Environment Variables Reference*.

Define a synchronization method

1. Select the imported replica to be synchronized.

2. Click **Tools→Import→Remote**.

3. In the **Import Remote** dialog box, click **Import Remote Options** in the lower right corner.

4. In the **Import Remote Options** dialog box, click the **Advanced** tab.

5. In the **Synchronization/Notification Options** pane, select the type of synchronization and/or notification required. For option descriptions and requirements, see *Data synchronization options*.

Default synchronization behavior

If none of the options are set in the **Import Remote Options** dialog box, the default synchronization behavior for imported replicas is as follows:

- If the object is being imported for the first time, the default synchronization method is through batch mode using the **data_sync** utility. There is no notification.

- If the object was previously imported, the option settings that were last set are used.

Visualization data synchronization

In a Multi-Site Collaboration environment, you can develop components and sub-assemblies at multiple sites while the entire assembly is configured at a single site. For collaborative design tasks, such as design reviews, you view visualization data for your parts, components, and assemblies. For the collaborative design tasks to be effective, the visualization data created during the collaborative tasks must be replicated and synchronized, along with the original (derived) visualization data. Because the visualization data is usually for a sub-assembly or assembly, Multi-Site Collaboration manages direct model (JT) files, 2D drawings, images, and documents associated to an item revision. Authored visualization data is data that references the derived visualization data in order to create higher level visualization functionality, and this data is authored directly by the core visualization tools. Examples of authored visualization data include PLM XML structure captures, markups, product views, sessions, and work instructions. When you create a visualization session or mark ups of an existing visualization session, the visualization data is replicated and synchronized to the sites where the original (derived) visualization is replicated.

The synchronization includes visualization datasets directly related to item revisions, and visualization datasets related to datasets related to item revisions. If a visualization dataset is related to a CAD dataset (and not the item revision) and the site intent is visualization synchronization, this visualization dataset is not replicated and synchronized because its parent item is not replicated and synchronized.

Synchronize visualization data only

You use the **-OnlyVIS** argument with the **data_sync** utility to synchronize only the visualization datasets that is related to a replicated item revision with status. This argument requires that you include arguments for the relations that you want to

include and may include a date argument (**-since**) which causes synchronization of visualization data that has been modified after the date. For example:

```
data_sync -u=infodba -p=infodba -g=dba -OnlyVIS -include=IMAN_Rendering
-include=IMAN_Manifestation -since=2005-01-01:01:01
-site=cologneTdsm -sync -update -report=report.lst
```

For more information on the **data_sync** utility, see the *Utilities Reference*.

Delayed synchronization of bulk data

It may be necessary to synchronize bulk data at a later time for performance reasons as the bulk data can be large and require a lot of bandwidth. During peak usage periods, using the **-exclude_files** option of the **data_sync** utility synchronizes only the metadata and leaves the associated files (bulk data) unchanged on the remote sites. When more bandwidth becomes available, you use **ImanFile** as a value for the **-class** option which causes the **data_sync** utility to synchronize only the bulk data for the item.

Synchronize bulk data only

Use the following command line entry to synchronize bulk data copied to **Site1** and output a report to the **report.lst** file:

```
data_sync -class=ImanFile -site=Site1 -sync -update -report=report.lst
```

Use the following command line entry to synchronize bulk data with the dataset names in the **ListOfDataSets.txt** file:

```
data_sync -filename="C:\ListOfDatasets.txt" -classoffile=ImanFile
   -site=Site1 -sync -update -report=report.lst
```

For more information on the **data_sync** utility, see the *Utilities Reference*.

Enabling automatic synchronization

The user that replicates an imported object can specify that the replica be synchronized automatically when the master object is modified. The replica is synchronized automatically using the Multi-Site Collaboration automatic synchronization option. This results is an efficient and evenly distributed synchronization process and replicas are updated within minutes after the master copy is modified.

Process daemons

Automatic synchronization requires the **subscriptionmgrd** and **actionmgrd** process daemons to be enabled at the owning site. These process daemons must be logged on to the owning site database.

For information on enabling these daemons and setting their respective preferences, see the *Subscription Monitor Guide*. You can also access the same help information by running the daemons as a utility with the **-help** switch.

Preferences

Automatic synchronization requires the **TC_subscription** site preference in the preference XML file to be set to **ON** at the owning site. This preference enables the subscription functionality upon which automatic synchronization depends.

E-mail notification

Notification of either type of synchronization requires the following conditions.

- The **subscriptionmgrd** and **actionmgrd** process daemons must be enabled at the importing site. These process daemons must be logged on to their respective sites database.

  For information on enabling these daemons, see the *Subscription Monitor Guide*. You can also access the same help information by running the daemons as a utility with the **-help** switch.

- Both the owning site and the replicating site must set the **TC_subscription** site preference in the preference XML file to **ON**.

- System e-mail must be enabled at the replicating site using the **Mail_server_name** site preference.

Chapter

6 System administration

Chapter

6 System administration

This information assumes that you have installed and configured Multi-Site Collaboration according to the instructions and guidelines in the installation guide for your platform and you are familiar with the following basic system administration concepts and features:

- Changing preference settings with the **preferences_manager** utility
- Using utilities
- Using Access Manager (AM)

For information about changing preference settings, see the *Preferences and Environment Variables Reference*. For information about using Teamcenter utilities, see the *Utilities Reference*. For information about using Access Manager, see the *Access Manager Guide*.

The Multi-Site Assistant tool is available for download from the Global Technical Access Center (GTAC). This tool has the following capabilities that can help you administer your Multi-Site environment:

- Site ownership analysis
  - Identifies all inconsistent item and item revision level ownership that impact end user operations
  - Identifies all inconsistencies between items and item export records that impact end user operations
  - (Optional) Analyzes objects below the item revision, such as datasets and occurrences, for inconsistent ownership
  - Suggests resolution commands required to fix the inconsistencies
- Site consolidation profile
  - Identifies potential target sites where you can consolidate a specific source site
  - Tracks the progress of consolidation during the pre-consolidation phase by providing the percent consolidated for a specific source and target site
  - Identifies data conflicts between source and target sites, such as duplicate Item IDs that must be addressed prior to consolidation
- Site metrics

  Provides site statistics and identifies the sharing patterns

If you have a Web Key account you can download this tool and its associated guide from the following locations:

http://ftp.ugs.com/teamcenter/MP/windows/tools/
http://ftp.ugs.com/teamcenter/MP/solaris/tools/
http://ftp.ugs.com/teamcenter/MP/linux64/tools/
http://ftp.ugs.com/teamcenter/MP/hp/tools/
http://ftp.ugs.com/teamcenter/MP/aix/tools/

Best practices

The following topics provide information about common issues the system administrators in a distributed environment must consider, and suggest strategies for successfully implementing and maintaining a Multi-Site Collaboration network:

- Object naming conventions
- Networking
- Security
- Site compatibility
- POM transmit schema files
- Database backup
- Multi-Site Collaboration accessors
- Remote checkin checkout

Object naming conventions

Some portions of an enterprise may have unique identifiers and they are typically represented in Teamcenter by their item IDs. Unique item IDs are enforced in a single database, but uniqueness cannot be enforced in a distributed environment. Multi-Site Collaboration assists you in detecting naming conflicts during publication of Items. At publication time, Multi-Site Collaboration searches the ODS for duplicate Item IDs and refuses to publish an item with a duplicate ID.

Some enterprises use temporary IDs. You can change these temporary IDs to real IDs later (either before or during a release process). These temporary IDs should be handled in the same manner as permanent IDs. If the same temporary ID is used at multiple sites, it prevents items from being shared among those sites. Siemens PLM Software recommends using a corporate-wide naming convention for temporary IDs, just as Siemens PLM Software recommends one for permanent IDs, in order to avoid conflicts of this type.

Networking

Multi-Site Collaboration is designed to optimize performance for local users at each site in the Multi-Site Collaboration network. During the design, an attempt was made to minimize the need for high-speed networks among various sites on the network. Siemens PLM Software does not have specific network requirements for

each of your sites, but recommends the highest performance network you can deploy, though Multi-Site Collaboration should function well over a lower-speed line. The network performance depends on how much use you make of the distributed system, especially with regards to synchronization, which is discussed later.

Multi-Site Collaboration makes use of the network for three functions:

- Searching for objects
- Publishing objects
- Retrieving objects

For each of these functions, there is not a considerable amount of network traffic. Network traffic is kept to a minimum by doing most of the work on the server side of the request and sending data across the network in large blocks. Once the data is received on the client side, no further use of the network is required. Some queries can use a lot of the network bandwidth (for example, should a user issue a query and ask for a large number of objects to be displayed), but most should use modest amounts. However, importing an object from a remote site is the most common task that requires a considerable amount of network bandwidth. When an object is imported, the metadata, which represents the object and (optionally) the bulk data from volumes, is transferred to the requesting site. The bulk data is usually much larger than the metadata and has more of an impact on network performance. The speed of the network you require is largely dependent on how many times you must import or reimport an object.

Security

In the early stages of planning a Multi-Site Collaboration network, it is important to identify the security issues that must be addressed and implemented. For detailed information on certain aspects of implementing security for a Multi-Site Collaboration environment that the system administrator of each site must know how to set up and implement, see *Planning and setup*.

Protecting shared data using AM rules

Once each system administrator has determined the site configuration, which sites are able to access data from which other sites must be decided. Typically, the system administrator sets up the sites so that all of the other sites in the enterprise can access data created by each of the other sites. However, this is not a requirement. The system administrator is able to set up rules on the data that dictate which sites can and cannot access various objects based on their types, their release status, or some other object property.

> **Warning**
>
> The system administrator must be careful about these access rules to ensure that they are consistent throughout the enterprise.

For example, if the system administrator sets up a rule stating that all items of type X are importable by sites 1, 2, and 3, the system administrator must ensure that associated objects such as the item master forms, requirements documents, and the specifications are all importable by those sites. If not, the export of those objects to

those sites fails because certain pieces of an item are required to be imported when importing an item.

For additional information about rules-based object protection and AM, see the *Access Manager Guide*.

When a user imports an object, the user has the option of importing a read-only copy of the object or of taking site ownership of that object. Taking site ownership of an object is a significant event and requires **TRANSFER_IN (i)** privilege for the importing site. If the requesting site does not have **TRANSFER_IN (i)** privileges, then object ownership cannot be transferred.

Two types of privileges are required for an object to be exported to a particular site:

- **EXPORT(X)** (for the exporting user at the owning site
- **IMPORT (I)** for the importing site

The user who exports the object must have **EXPORT(X)** privilege in order to export an object. Conversely, the site that imports the object at the destination site must have **IMPORT (I)** privilege to import the object. Note that the **IMPORT (I)** privilege is associated with the importing site and not the importing user. Both of these privileges are enforced from the site where the object is owned. To enforce the protections on an object, Siemens PLM Software has placed a restriction in the system so that users can only export an object from the site that owns the object. This provides better access control on the object and also ensures that the requesting user is always receiving the latest version of the objects, that is, not a copy of a copy.

ODS security

For those administrators managing ODS sites, a different type of security setup is required in order to control access to publication records. AM rules must be set up to key on the attributes of a publication record and prevent read access for sites that are unauthorized to access certain publication records. For example, a rule can be defined for publication records such as **If owning site is Ohio, then only sites Michigan and Illinois have READ access to the publication record**.

Note

You can use the **Any Attribute of Any Class** feature in the Access Manager to define Access Manager rules for publication records. For example, you can create a rule that states all publication records owned by site ID 123456789 can be accessed only by sites listed in the Named ACL **ods_security_acl**:

```
Has Class(PublicationRecord)
Has Attribute(PublicationRecord:pubr_owning_site=123456789)->
 ods_security_acl
```

A user at a remote site that does not have **READ** access to a certain publication record sees the *** **ACCESS DENIED** *** message for each publication record the site is unauthorized to access. However, the ODS system administrator can suppress this message, and to the user, the publication record does not exist at all. To suppress the ***ACCESS DENIED*** message at the ODS site, set the **ODS_suppress_pubrec_if_no_access** site preference to **TRUE**.

Controlling remote import capability

The default rule tree grants remote import capability to every user in the local database. Each local user can perform remote import from all remote sites. Because the remote import operation places a substantial load on the local system, network, and the remote site, it may be necessary to place some controls so that only those local users specifically granted the privilege can perform a remote import operation.

To do this, it is necessary to change the default rule tree.

There are two methods of controlling the remote import capability.

- The first, and easier method, controls remote import capability from all remote sites for each local user and/or group, that is, all-remote-sites-or-nothing.

- The second method provides a more granular control as it defines access rules for each remote site for each local user and/or group.

In both cases, the basic mechanism involves controlling access to site objects in the **POM_imc** class. By revoking the **IMPORT(I)** privilege on a site object for a given user, you effectively prevent the user from importing from that site. You can also prevent remote import with transfer of ownership by revoking the **TRANSFER_IN (i)** privilege from the site object.

Set up all-remote-sites-or-nothing controls

To set up all-remote-sites-or-nothing controls based on when you first configured your database (if you are not sure, see the following tips) go to:

- *Create named ACLs for each remote site*

- *Create a rule tree entry for each site*

Create named ACLs for each remote site

If you want to implement controls on remote import on a per site basis, you must first implement the all-remote-sites-or-nothing approach as previously described, but without adding entries for individual user or groups in the remote import named ACL.

1. For each remote site that you want to control access to, create a named ACL that is used as a site-specific version of the remote import name ACL described above.

   For example, create named ACL **Site1_Remote_Import** for Site1, **Site2_Remote_Import** for Site2, and so on.

2. For each site-specific named ACL, add entries for individual users and/or groups to grant **IMPORT** and/or **TRANSFER_IN** as appropriate.

Create a rule tree entry for each site

At this point, regardless of when you first configured your site, your rule tree should have the **Has Class(POM_imc)→Remote Import** entry.

1. Click **Has Class(POM_imc)→Remote Import** in the rule tree and modify it. **Type Has Class(POM_imc)**.

2. Click the new entry and add an entry for each remote site as follows (Assume that the site ID for Site1 is 111111111, and so on):

```
Has Class(POM_imc)
Has Attribute(POM_imc:site_id=111111111)-
>Site1_Remote_Import
Has Attribute(POM_imc:site_id=222222222)-
>Site2_Remote_Import
Has Attribute(POM_imc:site_id=333333333)-
>Site3_Remote_Import
 Has Class(POM_imc)→Remote Import
```

Note

The last entry is a catch-all entry for all other sites and may be deleted or added as desired.

Site compatibility

To share data among various sites, site configurations must be compatible. Certain system information must be set up consistently in order to import data and work with the data once it has been imported. There are certain aspects of a site that must be identical and certain aspects that must only be compatible. The **database_verify** utility is used to compare any two Multi-Site Collaboration sites for compatibility.

The following site data elements must be identical among all Multi-Site Collaboration sites (or one at least be a subset of one other) in order for data exchange to be possible:

- Types (for items, relations and all other types)
- Dataset types
- Form types
- Units of measure
- Tools
- Note types

The schemas can be extended but they must be compatible. Although for an object to be imported into a site, the class that is being imported must be defined with a compatible set of attributes at the receiving site. Compatible attributes are attributes that are defined at both sites with the same name and type and the receiving site attributes must be a superset of the sending site.

For example, if Class1 is defined at two sites and the definition of Class1 contains the following attributes at the sending site:

- **attr1** – integer
- **attr2** – integer array (size 3)
- **attr3** – string[32]

The definition of Class1 at the receiving site must contain those same attributes with the same type and size specifications. The receiving site can have additional attributes defined, but it must have all the attributes defined by the sending site.

In a Multi-Site Collaboration environment, a warning appears when a 128-byte site sends an item ID or name longer than 32 bytes to a 32-byte site. You should upgrade your sites to 128-byte functionality to avoid this issue.

POM transmit schema files

When an object is transferred from one site to another, a POM transmit schema file is required. This file must be regenerated and stored in the **$POM_TRANSMIT_DIR** directory before using Multi-Site Collaboration. Whenever the schema at a site is changed, the POM transmit schema file must be regenerated and distributed to all sites in the network. The **install** utility can be used to regenerate this file. For additional information about regenerating the POM transmit schema file, see *Utilities Reference*.

Set POM transmit variables

Multi-Site Collaboration creates a POM transmit schema file. Previous versions use a **$** character in the filename. The current version uses a **-** character in its place. To make POM transmit schema files compatible with previous versions of Teamcenter and Multi-Site Collaboration, you must set a POM transmit variable on the current system and the previous one. Make sure that the variables are set opposite of one another on the current and previous systems, to ensure that they generate and locate the right files names.

POM_TRANSMIT_NEW_NAMES

To configure a site that is running a previous version of Multi-Site Collaboration, you can generate a POM transmit schema file that is compatible with a current version by setting the **POM_TRANSMIT_NEW _NAMES** variable. If you set the variable to **ON**, the POM transmit schema file is created with the **-** character. If it is set to **OFF**, the file is created with a **$** character.

POM_TRANSMIT_OLD_NAMES

To configure a site that is running a current version of Multi-Site Collaboration, you may generate a POM transmit schema file that is compatible with a previous version by setting the **POM_TRANSMIT_OLD _NAMES** variable. If you set the variable to **ON**, the POM transmit schema file is created with the **$** character. If it is set to **OFF**, the file is created with a **-** character.

For more information about the POM transmit schema files, see the .

Database backup

Multi-Site Collaboration enables object sharing among multiple databases; it creates one large virtual logical database throughout an enterprise. Therefore, it is extremely important that all databases on the Multi-Site Collaboration network are backed up regularly and are secure.

The first technique would be to make a full backup of all of your databases on a regular basis. All databases can be backed up using a time stamp through database backup procedures. Using a time stamp helps you to restore all your databases to a certain point in time should one of them crash. When backing up your databases, it is important to ensure that you back up all of your volumes as well as the metadata.

When using backups to maintain your databases, it is a good practice to maintain database transaction logging so that databases can be rolled forward to the last committed transaction should one of them go down.

Should you be forced to restore a database from a backup tape, it is important to verify that no objects have been restored whose ownership had been transferred to another site between the time of the last backup and the date of the crash. Should such an event take place, you must reimport the object from the appropriate site. If this database had taken ownership of an object from another site, ownership of that object will have been lost from the network. The object can be restored from an export file if one still exists or has to be corrected manually as a last resort.

Multi-Site Collaboration accessors

Access Manager (AM) uses the following two special accessors with Multi-Site Collaboration:

- **Site=***site_id*
- **Remote Site**

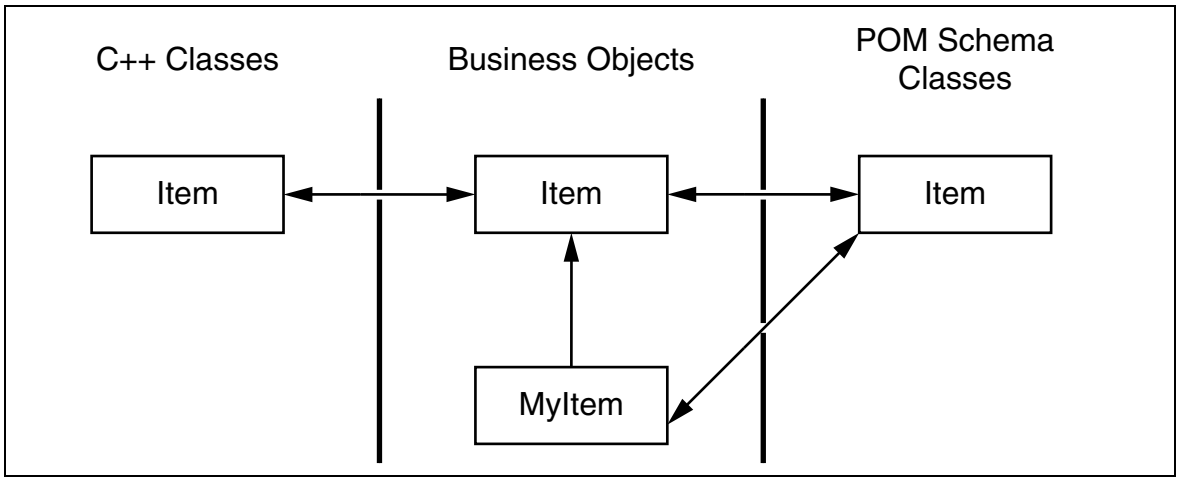
The accessors are used to identify particular sites when adding a new rule to the AM rule tree. The first accessor is used to specify a particular site by its unique site ID; the second accessor is similar to world, it represents all other sites.

For additional information about adding rules to the AM rule tree, see the *Access Manager Guide*.

Compatibility with earlier versions

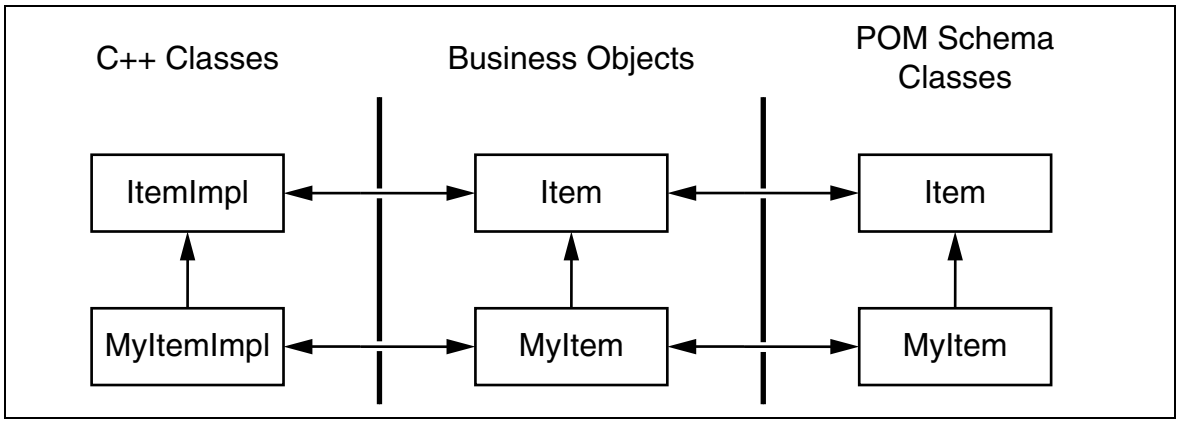
Any custom attributes on forms in earlier Teamcenter versions are not imported into the primary class object in Teamcenter 8.3. The type and class mapping is maintained to allow the data to be passed between the systems.

The following figure shows the **MyItem** subtype for the parent **Item** class.



MyItem subtype from parent Item class

The following figure shows how this is represented in Teamcenter 8.3 after the mapping to the **MyItem** subclass of the parent **Item** class.



MyItem subclass from parent Item class

Any additional attributes on a subtype master form are lost when you import the object from an earlier version of Teamcenter. Also any attributes on a subclass are lost when you export the object to an earlier Teamcenter version. Only attributes stored on the primary business object are transferred.

For information about converting secondary objects to primary objects, see the *Business Modeler IDE Guide*.

You can add attributes to forms in Teamcenter 8.3 like in earlier versions. Adding the attributes from forms in an earlier version of Teamcenter avoids data loss when transferring data back to the earlier version.

During import, Teamcenter uses the persistent object model (POM) to create stubs for all referenced instances on the object. These are augmented stubs, which include the **object_type** information. Teamcenter uses this to determine if the object has been mapped from a previous version, and if so, the class of the stub. Teamcenter determines the **class_name** value from the type value.

During an import, if the data related to a subtype from an earlier version is missing, and a stub must be created for the object, the stub references the subclass of the object present in Teamcenter and not the class on the source site.

Existing stub data from the migrated system contains references to the subtype of **Class**. When you import this business object, it is imported into the subclass and not the parent class as designated by the stub.

Generate a dataset mapping file

You must create a dataset mapping file to allow data transfers between Teamcenter 8.3 and earlier versions. If you do not generate a dataset mapping file for a site, Multi-Site assumes that no type to class conversion is required between the participating sites. The **database_verify** utility can generate mapping files for connected sites. To do this, use the **database_verify** utility to create a **_TCYTPES_SITE_** file for each remote site. Use the dataset mapping file to provide mapping information to the **item_import** and **item_export** utilities.

If a site is not connected (offline), you must generate a list of types manually using the **list_types** utility You generate a list of types file at the offline site and supply this file as input to the **database_verify** utility to generate the dataset mapping file. You must regenerate the **_TCYTPES_SITE_** file for a site any time there is a change to the POM transmit file for either site.

Note

The utilities used in the following examples required Teamcenter administrator privileges. If you are logged on to the Teamcenter host using the same credentials as a Teamcenter administrator, you can omit the **-u** and **-p** arguments.

- To generate mapping files for all sites defined in the local Teamcenter database, from a command prompt, type:

  ```
  database_verify -u=admin-user -p=adminuser-password -site=ALL
  ```

  The utility creates a file for each site named **_TCYTPES_SITE**_*site-id*. *site-id* is the **Site ID** value entered when you created the site in the Organization application.

  For information about creating sites, see the *Organization Guide*.

- To generate a mapping file for a specific site, type:

  ```
  database_verify -u=admin-user -p=adminuser-password -site=site-id
  ```

- If you want to create a new mapping file for a site that is offline, you must have a types file for the site available.

  1. To create the types file, at the offline site, type:

     ```
     list_types
     -u=admin-user -p=adminuser-password -outfile=types-file-name
     ```

  2. Transfer the types file to the host of the other Multi-Site Collaboration site.

  3. To create the mapping file, type:

     ```
     database_verify
     -u=admin-user -p=adminuser-password -site=site-id -offline
     -filename=types-file-name
     ```

Remote checkin and checkout administration

If your users replicate NX assemblies with arrangements or use remote checkin or checkout on such assemblies, include the following relations in the **TC_relation_required_on_transfer** and **TC_relation_required_on_export** site preferences:

- **TC_Arrangement**
- **TC_DefaultArrangement**
- **TC_BaseArrangementAnchor**

This eliminates the requirement to select these relationships during the import, export, checkin, or checkout action.

Item ID consolidation

The following process flow and utilities are designed to help you reconcile duplicate items between multiple databases. This is required prior to sharing information across Multi-Site Collaboration sites. The system administrator runs the utilities. For additional information about running the **item_rename** and **item_relink** utilities, see the *Utilities Reference*.

Item ID process flow

The algorithm to eliminate duplicate items across sites can be done in three major steps:

1. Identify the ownership of all part numbers. This step is performed manually.

   Each site must examine its database and determine the ownership of all parts in the database.

   The parts that the site owns is the sites master data. The parts that the site does not own are duplicates that are replaced by replicas.

2. Publish master data to the ODS. This step is performed manually.

   The master data must be published if it is to be shared.

3. Replace duplicates with replicas.

   Follow these steps in the sequence documented in the following table to replace duplicates. This process should only be performed by experienced users with system administration and Multi-Site Collaboration expertise.

Process	**Action**
Manual process	Move the duplicate items to a replacement folder.
Run **item_rename**	Run the **item_rename** utility to change duplicate item IDs.
Manual process	Create the replica data objects. • Access the Multi-Site Collaboration network environment and use the **Remote Import** command to import the master items from the owning site. • Import all the NX items that are identified as master copies of the duplicates. This process creates the replica data objects with the original Item ID naming convention.
Run the **item_relink**	Run the **item_relink** utility to search for every link that is connected to the duplicate data objects. Replace this link with the link that is connected to its corresponding replica data object.

Process	Action
Run **ugmanager_refile**	Run the **ugmanager_refile** utility to update the top-level NX assemblies that referenced all of the parts that were processed as replica data objects. For information about the **ugmanager_refile** utility, see Teamcenter Integration for NX in the NX online help collection.
Manual process	Delete the duplicates data objects.

Handlers

If your site is using both Multi-Site Collaboration and workflow, the following Enterprise Process Modeling (EPM) handlers are used to publish, unpublish, or send (export) objects to various sites directly from workflow jobs:

- **EPM-publish-target-objects**
- **EPM-send-target-objects**
- **EPM-unpublish-target-objects**

  Note

  Do not set the **EPM-unpublish-target-objects** handler on the **Perform** action or any other action than can be called multiple times. Place this handler on an action which is called only once, such as **Start**, **Complete**, or **Undo**. For additional information about workflow, see the *Workflow Viewer Guide*.

  For additional information about workflow handlers, including how to add these handlers in your workflow jobs, see the *Workflow Designer Guide*.

Utilities

The following utilities support Multi-Site Collaboration functionality:

- **database_verify**
- **export_recovery**
- **data_share**
- **data_sync**
- **item_relink**
- **item_rename**

For additional information about using utilities, see the *Utilities Reference*.

Distributing system administration data

Distributed system administration functionality simplifies the task of administering multiple sites. It allows you to duplicate the data using the **dsa_util** utility, rather than logging into each remote site and manually recreating the data. For example, after you have defined new users and groups at site A, you can create these same entries at site B, site C, and site D using the utility. You do not have to log in to the remote sites.

Caution

Do not use the **dsa_util** utility to import data model objects managed by the Business Modeler IDE. This can corrupt those objects. Use the Business Modeler IDE to manage all custom data model objects.

For information about working with data model files using the Business Modeler IDE, see the *Business Modeler IDE Guide*.

Distributing Teamcenter system administration data differs from distributing Teamcenter product data. Exchanging system administration data involves the distribution of *system objects* such as Teamcenter users and groups. Exchanging product data involves importing and exporting workspace objects such as datasets.

Creating distributed system administration data differs from creating distributed product data. Distributing system administration data creates new system objects at the target remote site. This cloned data is owned by the remote target site, rather than transferring ownership from the originating site. Distributing product data replicates existing workspace objects, which are shared between sites.

The following are example scenarios in which distributing system administration data is useful:

- Replica ownership depends on whether the owning group/user at the owning site is defined at the importing site. The only way to preserve the owning group/user across sites is by defining the owning group/user at all importing sites. Without using distributed system administration functionality, all persons, users and groups would have to be manually created at all importing sites. Distributed system administration functionality allows you to create all the necessary data at one site, then propagate the information to all affiliated sites with a single utility command.

- If you manage various sites that do not share data using Multi-Site Collaboration, you can still use distributed system administration functionality to manage your various sites. If, for example, all your sites follow the same basic business practices, you can use distributed system administration functionality to distribute, release statuses, revisions rules, and so forth.

Security controls

Site-level security of distributing system administration data is enforced using the **IDSM_dsa_sites_permitted_to_push_admin_data** site preference. Each local site must define this preference in its site preference XML file. Set this preference by listing all the remote sites to be allowed to distribute system class information to the local site.

If this preference is not defined, no remote site is allowed to distribute any system classes to the local site.

Classes, instances, and attributes

Understanding POM classes in general and system classes in particular allows you to optimize distributed system administration functionality. This topic provides a brief overview of Teamcenter classes, instances and attributes. For additional information about classes and attributes, see the *Business Modeler IDE Guide*.

Overview

Distributing system administration data involves distributing system classes. These are database classes that include **User**, **Group**, and **Person**.

The database stores the Teamcenter data in tables. Each table is populated by rows, and each row represents an entry in the table. For example, there is an database table that contains an entry (or row) for each person defined in the Teamcenter database. Each row consists of several columns. Each column contains specific information about the person. For example, there is a column for the person name, a column for the e-mail address, and so forth.

From a Teamcenter perspective, it is easiest to consider the database tables as a class of objects. Because multiple tables typically represent a single class, it simplest to consider a set of related tables as a class. For example, there is a class for storing **User** information, a class for storing **Group** information, and so forth.

Note

Class names are not case sensitive. Siemens PLM Software recommends you use mixed case when specifying class names for easier readability. For example, specify **GroupMember**, rather than **groupmember**.

The specific entry (or row) in a table is referred to as an *instance of a class*. When you define a particular person in Teamcenter, you are creating an instance of the **Person** class. Within the database, you are actually creating rows in several tables at once, which is why it is best to view these entries in terms of classes and instances. An instance is referred to as an object because it represents something tangible in the database.

Each class contains one or more attributes, which correspond to the columns in the table. At the Teamcenter application level, attributes are referred to as properties. However, when dealing with system classes, it is better to refer to them as attributes because related tools such as the **taxonomy** utility use this term. Each attribute contains a name, one or more values, and a set of characteristics.

For example, the **Person** class might have the following attributes:

- **user_name**
- **PA1**
- **PA2**
- **PA3**
- **PA4**

The **user_name** attribute corresponds to the **Person Name** label that displays in the interface. The **PA1** attribute corresponds to the **Street Address** label. The **PA2** attribute corresponds to the **City** label, and so forth.

Distributing class data

To distribute information about a particular person, you identify that person with the **user_name** attribute.

For example, to distribute information about Joe Smith, instruct the utility to distribute information on class **Person** where the **user_name** attribute is equal to **Joe Smith**. Perform a visual check of the taxonomy report to confirm that **user_name** does have a unique value with the **Person** class. Such a unique value is known as a key attribute because it can be used to identify specific unique instances of a class. It is important to know the key attributes of a class when using the **dsa_util** utility to distribute system administration data.

In another example, you could distribute person information by instructing the utility to distribute information on **Person** class where the **PA2** attribute (equal to **City**) is equal to **Los Angeles**. This distributes all persons defined in the local database who live in Los Angeles.

Using the dsa_util utility

Distributing system administration data involves the distribution of system classes. The **dsa_util** utility is used to distribute these system classes.

For consistency with other Teamcenter applications, the syntax of the command line switches, especially those related to class and attribute names, match those in the taxonomy report. This report is generated by the **taxonomy** utility. For example, when distributing groups, the class name is **group** and an argument for the attribute **name** is **-name=engineering**.

> **Note**
>
> Some exceptions to these naming conventions are made in cases where the POM class name (the name displayed in the taxonomy report) is not descriptive. For example, the **POM_imc** class does not clearly indicate that this class is used to store site information. In such cases, both the POM class name and the descriptive alias class name are both accepted. See example 4.

dsa_util utility behavior

Before using the **dsa_util** utility, it is important to understand the following aspects of the utility's behavior:

- The system information to be distributed to other sites must exist at the local site where the **dsa_util** utility is being run. For example, to distribute information about Joe Smith, all **Person**, **User**, and **Group** information about Joe Smith must already exist in the local database.

- The information to be distributed is always written to a text file that is then sent to the destination sites for processing. While this text file is normally invisible to the user, the **dsa_util** utility enables you to work directly with the text file. You can edit the text file using your favorite text editor, then instruct the **dsa_util** utility to process the text file. For additional information, see *Text file format*.

- The **dsa_util** utility employs dynamic command arguments based on the database-defined attributes of the class to be processed. For example, when distributing the **Person** class, you can use the **user_name** argument because the **Person** class has an attribute of that name, not because this argument is a fixed command argument of the utility.

For complete usage documentation of this utility, see the *Utilities Reference* manual or run **TC_BIN/dsa_util -h** to display the documentation from the command line.

Usage examples

The following examples illustrate typical usage of the **dsa_util** utility to distribute system objects. The examples all assume the system objects already exist at the local site. The system objects may or may not currently exist at the target sites.

For information on formatting the text file generated by the **dsa_util** utility, see *Text file format*.

For information on reporting results of the requested operations, see *Generating reports*.

Example 1

To distribute person information to a remote site:

```
dsa_util -f=distribute -class=Person -user_name="Joe Smith" -site=site1
```

This command creates a new **Person** object at the destination sites, if the person does not already exist, or updates the existing object. The instance of the **Person** class does not have to be a replica, but must be owned by the destination site.

This is the default behavior when distributing system objects.

Note

user_name is an attribute of the **Person** class, not a fixed **dsa_util** argument.

Example 2

To distribute user information to selected remote sites:

```
dsa_util -f=dist -class=User -user_name=joe -site=site1 -site=site2
```

This command requires the creation or update of other system objects such as **Person** and **Group**. While the references **Person**, **Group**, and **Role** objects can be sent using a separate argument, this is not required. Typically, when sending a class such as **User**, referenced classes are created or updated automatically. In this case, **Person** and **Group** objects are created or updated.

Example 3

To distribute information on multiple users whose user IDs are contained in a **users.ls** text file:

```
dsa_util -f=dist -class=User -user_id_listfile=users.ls -site=site1
```

The use of the **user_id_listfile** argument indicates that the text file named **users.ls** contains a list of IDs to be distributed, one user ID per line. Typically, the **_listfile**

postfix should be appended to the attribute name if the attribute values are to be supplied using a text file.

Example 4

To distribute site information to multiple remote sites:

```
dsa_util -f=dist -class=Site -site_id=123456789 -site=Site1 -site=Site2
```

The alias class name **Site** is used instead of **POM_imc** which is the POM class name. **POM_imc** is also be accepted as the class name.

Text file format

System administration data can be distributed between sites much like product data. System class information is distributed by using the **dsa_util** utility to define the information to be distributed. The utility then outputs the system class information into a text file in Extended Markup Language (XML) format.

You can view and modify this text file before distributing the system class information. You can also use this text file as direct input to the utility. This allows the distribution of system classes between sites that are not using Multi-Site Collaboration.

Usage examples

Example 1

To output system information on all users defined at the local site to a text file:

```
dsa_util -f=export -class=User -user_id=* -filename=users_info.xml
```

This command creates the **users_info.xml** text file and writes all user information in the database into the text file. The file can then be edited using any text editor, if required. Siemens PLM Software recommends you use XML editors that are available off-the-shelf. You can use the text file in the following ways:

- The local system administrator can use the text file as input to the **dsa_util** utility for distribution to other sites.
- Copy the file onto a CD-ROM and send the text file to a site not configured for Multi-Site Collaboration.

Example 2

To output multiple system classes to a single text file:

```
dsa_util -f=export -class=User -class=Site -filename=sys_info.xml
```

This command outputs all instances of the **User** and **POM_imc** classes.

Example 3

To read system class information from a CD-ROM and update the local database:

```
dsa_util -f=import -filename=/cd_device/sys_info.xml
```

This command reads the information from the **users_info.xml** class in the CD-ROM and updates the local database.

Example 4

To use a given text file for distribution to other sites:

```
dsa_util -f=import -filename=/cd_device/sys_info.xml
```

This command distributes system classes defined in the **system_info.xml** text file. Typically this command is used after the local system administrator has output system administration data to a text file, made necessary manual edits and is now ready to distribute the information to other sites.

Generating reports

System class information is distributed by using the **dsa_util** utility to define the information to be distributed. The utility then outputs the system class information into a text file. The utility can also generate reports that display the results of requested operations.

Distributed system administration functionality provides two types of reports: a local report and a remote report. To fully understand the significance of these reports, you must first understand the client/server architecture of this functionality.

The local report is produced by the client process and provides information regarding operations performed by the client process. When distributing system objects, the client process exports data from the database into an XML-formatted text file and sends the file to remote sites. The local report is primarily about export and send operations. The local report shows which system objects are exported and the status for each object.

The remote report is generated by the server process, the IDSM server. Because the server receives and imports information into the remote database, the remote report is primarily about data reception and import operations.

The remote report is delivered through e-mail. The user that initiates the distribution from the master site can receive a copy of the remote report by specifying an e-mail address when running the **dsa_util** utility. The system administrator of the remote database or the receiving site can receive the report by defining the **IDSM_dsa_notification_email_address** site preference to specify an e-mail address or a set of e-mail addresses where the report should be sent to.

Usage examples

The following examples illustrate how to specify reports using the **dsa_util** utility:

Example 1

To distribute a list of users and generate a local report:

```
dsa_util -f=distribute -class=User -user_id_list=users.ls -site=site1
-report=myreport.txt
```

This command generates a report in the **myreport.txt** text file that lists the user IDs processed and the status of each.

Example 2

To distribute a list of users, display a report on the window and generate a report from the destination site:

```
dsa_util -f=distribute -class=User -user_id_list=us.ls -site=site1
-report=my_local_report.txt  -email=joe@xyz.com
```

This command generates a report in the **myreport.txt** text file that lists the user IDs processed and the status of each. The local report is output to the **my_local_report.txt** file in the user's current directory, and a report from Site 1 is generated and sent to **joe@xyz.com**.

The report at Site 1 can also be generated without the **-email** argument if Site 1 has defined a preference to always send a report to a designated e-mail address.

Controlled replication of structure context objects

Structured context objects (SCOs) represent a virtual product configurations. A project for the assembly can be spread across multiple sites. Because this information must be made available as quickly as possible to all participants, Multi-Site supports replicating these objects to sites participating on the assembly when it is released. To support this functionality, Multi-Site uses a Participating Sites form that contains a list of the sites associated with a project. This requires the assembly to be related to a project before it is released.

Note

This functionality is not intended for automatic object sharing or synchronizations. Updated or new objects are synchronized or replicated only when the SCO object is released.

If the assembly root object is not assigned to a project or does not have a related Participating Sites at the site where the assembly is released, Multi-Site cannot replicate the object and an error is logged by the replication utility.

If an assembly is related to multiple projects, you must create and maintain Participating Sites forms for each project at the top level assemblies owning site. This allows the assembly to be replicated to a consolidated list of site all projects that contain it.

The **EPM-release-and-replicate** workflow task handler triggers the replication in conjunction with the **CreateAssemblyPLMXML** translator that initiates the **validate_and_replicate_assembly** and **data_sync** utilities.

For information about the **EPM-release-and-replicate** workflow task handler, see the *Workflow Designer Guide*.

For information about enabling the **CreateAssemblyPLMXML** translator, see the *Dispatcher Server Translators Reference Guide*.

For information about data sharing utilities, see the *Utilities Reference*.

There are several site preferences you can use to control the replication behavior for SCO objects.

- By default, Multi-Site attaches the datasets using the dataset type, **IMAN_reference** relation, and **ConfiguredAssembly** named reference type. You can designate that SCO related datasets be attached to the assembly using a different, dataset type, relation, and/or named reference using the **TC_plmxml_sync_dataset** site preference.

- Multi-Site also uses the **DirectModelAssembly** dataset type, **relation**, and **ConfiguredAssembly** named reference type to determine the item revisions to import. You can set different values for processing the item revisions to import using the **TC_identify_plmxml_import_dataset** preference.

- Multi-Site imports item revisions as determined by the **TC_identify_plmxml_import_dataset** preference that are out-of-date. You can use the **TC_plmxml_import_item_filter** preference to prevent import of specific item revisions based on a **BOMLine** property.

For information about Multi-Site Collaboration preferences, see the *Preferences and Environment Variables Reference*.

If an attempt to replicate an assembly/SCO component fails, or the parsing of a PLM XML file to identify item revisions fails, you can resubmit the replication task using the Translation Admin Client at the participating site.

For information about using the Translation Admin Client, see the *Dispatcher Server Installation Guide*.

Chapter

7 Site information form

Chapter

7 Site information form

This site information form is provided to help you document your Multi-Site Collaboration network. Fill out one site information form for each site in your entire (that is, enterprise-wide) Multi-Site Collaboration network.

The site information form consists of two pages: Site Information and Preference Settings.

The following describes how to fill out the site information form.

Site information	Description
Site name, ID, and location	If this is an existing site, identify the site name and ID. If this is a new site, enter a short descriptive name for this site (for example, Design Center, Manufacturing, and so on). The site ID is generated automatically when you install Teamcenter at that site. Finally, record the geographical location of this site (for example, Detroit. Tokyo, and so on).
Working and/or ODS site	Indicate all that apply.

Network information	Description
LAN type	Enter the type of local area network (LAN) used to connect the clients to the database.
WAN	Enter the name of any other sites connected to this site through a dedicated wide area network (WAN).
Client types	Enter the types of client workstations or computers used at this site. You do not need to be exhaustive; general classifications are sufficient (for example, HP, UNIX, Windows, Intel PC, and so on).

Software information	Description
Database, Teamcenter, and Computer Aided Design (CAD)	Enter the database type and version, Teamcenter versions, and CAD types and versions used at this site.

Vital statistic	Description
Activities	Indicate all activities performed by this site.

Vital statistic	Description
# Users	Enter the total number of user accounts at this site. Retrieve this information by creating an Employee Summary report (select **Tools→Reports→Employee Report** in My Teamcenter).
# Existing items	Enter the total number of existing Items at this site. Retrieve this information by creating an item Summary report (select **Tools→Reports→Item Summary Report** in My Teamcenter).
# New items/Yr.	Enter an estimate of the total number of new items created yearly by this site.
Local schema upgrades?	Circle **YES** or **NO**. If you are certain that your enterprise always upgrades all POM schemas consistently throughout the entire enterprise, circle **NO**. If you are unsure, check **YES**. **Warning** All sites in a Multi-Site Collaboration network must have compatible Persistent Object Model (POM) schemas. If one or more of the sites has extended their schema by adding new classes and attributes, the rest of the participating sites must extend their schemas by adding those same classes and attributes.

Database object	Description
Node name and IP address	Enter the node name and internet protocol (IP) address of the database server hosting this site.
Host Type	Enter the database server platform type (for example, HP UNIX, Windows, and so forth) for this site.
Memory and disk space	Enter the amount of system RAM and hard drive space required for this database on the server.

IDSM daemon object	Description
Node name and IP address	Enter the node name and IP address of the computer or workstation running the IDSM daemon.

ODS daemon object	Description
Node name and IP address	Enter the node name, IP address, site name, and site ID of the default ODS database serving this site.

Site Information		
Site Name:	Working Site: Yes | No	ODS Site: Yes | No
Site ID:	Location:	
Networking		
LAN Type:	WAN Sites:	
Client Types:		
Software		
Teamcenter Version:	Database Type:	Database Version:
CAD Type/Version:	CAD Type/Version:	Other:
CAD Type/Version:	CAD Type/Version:	
Activities		
Engineering Design: Yes | No	Release Management: Yes | No	Other:
Manufacturing: Yes | No	Document Management: Yes | No	
Vital Statistics		
# Users:	# Existing Items:	# New Items/Year:
Local Schema Upgrades: Yes | No	Other:	
Database Server		
Node Name:	IP Address:	Host Type:
Server Memory (MB):	Server Disk Space (GB):	
IDSM Processes		
Node Name:	IP Address:	
ODS Processes		
Node Name:	IP Address:	
Site Name:	Site ID:	

SITE PREFERENCE SETTINGS	
Site Name:	
Working (IDSM) Site Preferences	
IDSM_permitted_sites=	ODS_site=
	ODS_searchable_sites=
IDSM_permitted_transfer_sites=	
	ODS_searchable_sites_excluded=
IDSM_restricted_sites=	
	TC_publishable_classes=
TC_transfer_area=	
ODS Site Preferences	
ODS_permitted_sites=	ODS_restricted_sites=
TC_ods_client_def_timeout=	TC_ods_client_initial_timeout=
Language Site Preference	
TC_master_locale_*site-name*=	
Replica Site Preferences	
TC_disallow_release_status_on_ replica=	TC_validate_stub_tickets=
TC_stub_dataset_files_after_ownership_ transfer=	TC_Populate_FSC_Server_Targets=
TC_always_exclude_dataset_files_on_ export=	TC_force_remote_sites_exclude_files=

Chapter

8 Custom configurations

Chapter

8 Custom configurations

You can customize your multiple site configuration on both UNIX and Windows servers.

Configuring multiple sites on a single server

You can configure Multi-Site Collaboration to support multiple sites on a single server. This configuration is available for UNIX servers and for Windows servers.

Siemens PLM Software recommends that you read the entire procedure before beginning the configuration for the appropriate server.

Configuring multiple sites on a UNIX server

You can configure Multi-Site Collaboration to support multiple sites on a single UNIX server.

Start with two fully installed sites of Teamcenter configured for Multi-Site Collaboration on a single server.

Set up multiple ODS daemons on a single server

Sun Solaris is used in the example; the sites are called **chicago** and **detroit**.

1. Gain root privileges.

2. Copy the **$TC_ROOT/bin/run_tc_ods** file to **$TC_ROOT/bin/run_tc_ods_chicago** and **$TC_ROOT/bin/run_tc_ods_detroit**.

3. Edit the **$TC_ROOT/bin/run_tc_ods_chicago** file by adding this argument to the ODS run line. Enter the following command:

```
nohup ${TC_ROOT}/bin/ods rpc_prog_number=536875585
   > ${TC_TMP_DIR}/ods$$.log 2>&1
```

4. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the Chicago site is installed.

5. Edit the **$TC_ROOT/bin/run_tc_ods_detroit** file by adding this argument to the ODS run line. Enter the following command:

```
nohup ${TC_ROOT}/bin/ods rpc_prog_number=536875584
   > ${TC_TMP_DIR}/ods$$.log 2>&1
```

6. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **detroit** site is installed.

7. Modify the **/etc/init.d/rc.ug.ods** file to call the two new scripts instead of the original.

8. Launch the script that runs the ODS daemon by entering the following commands:

```
su infodba -c "$TC_ROOT/bin/run_tc_ods &"
   su infodba -c "$TC_ROOT/bin/run_tc_ods_chicago &"
   su infodba -c "$TC_ROOT/bin/run_tc_ods_detroit &"
```

The **chicago** and **detroit** ODS daemons are now running on the same service.

Set up multiple IDSM daemons on a single server

Sun Solaris is used in the example; the sites are called **chicago** and **detroit**.

1. Gain root privileges.

2. Copy the *TC_ROOT***/bin/run_tc_idsm** file to *TC_ROOT***/bin/run_tc_idsm_chicago** and*TC_ROOT***/bin/run_tc_idsm_detroit**.

3. Edit the **run_tc_idsm_chicago** file by adding this argument to the exec IDSM line:

```
exec ${TC_ROOT}/bin/IDSM rpc_prog_number=536875586
```

4. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **chicago** site is installed.

5. Edit the **run_tc_idsm_detroit** file by adding this argument to the exec IDSM line:

```
exec ${TC_ROOT}/bin/IDSM rpc_prog_number=536875587
```

6. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **detroit** site is installed.

7. Edit the **/etc/inet/inetd.conf** file by commenting out the following line:

```
536875586/1 tli rpc/tcp nowait infodba ${TC_ROOT}/bin/run_tc_idsm run_tc_idsm
```

8. Edit the **/etc/inet/inetd.conf** file by adding the following lines:

```
536875586/1 tli rpc/tcp nowait infodba
   ${TC_ROOT}/bin/run_tc_idsm_chicago run_tc_idsm 536875587/
1 tli rpc/tcp nowait infodba ${TC_ROOT}/bin/run_tc_idsm_detroit
run_tc_idsm
```

9. The **inetd.conf** file is formatted differently depending on the platform you are using. Follow the format of the entry in the file.

   The **rc.ugs.ods** file is located in the following directories for the listed platforms:

Platform	Directory
AIX	**/etc**

Platform	Directory
HP-UX	**/sbin/init.d**
Solaris	**/etc/init.d**
Linux	**/etc/init.d**

For more information on platform differences, see the *Teamcenter 8.3 Release Bulletin*. The **chicago** and **detroit** IDSM daemons are now running on the same service.

Configuring site preferences

Configure site preferences on both sites using the site-specific RPC program number:

TC_*daemon-name_site-name*_**prog_number=***RPCProgramNumber*

This preference file is used by all sites accessing the remote sites.

For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

The default RPC program number for ODS is **536875585** and for IDSM it is **536875586**.

Additional ODS servers require unique RPC program numbers; use descending numbers beginning with the default number. Multiprocess ODS configuration requires an additional block of unused RPC program numbers for proper operation. The size of the block of numbers must be equal to the **ODS_multiprocess_max_subprocess_count** site preference value. The block of numbers must start immediately after the main ODS program number and be consecutive.

Additional IDSM servers require unique RPC program number; use ascending numbers beginning with the default number.

Note

The Noblenet Portmapper Service must be started before the IDSM and ODS service.

For example, add the following site preferences to the **Data Sharing.Multi-Site Collaboration** preference category on both sites:

```
TC_ods_chicago_prog_number=536875585
   TC_ods_detroit_prog_number=536875584
   TC_idsm_chicago_prog_number=536875586
   TC_idsm_detroit_prog_number=536875587
```

The following example illustrates multiprocess ODS in use at the Detroit site with an **ODS_multiprocess_max_subprocess_count** value of 10. Note the block of ten unused program numbers between **chicago** and **detroit**.

```
TC_ods_chicago_prog_number=536875585
   TC_ods_detroit_prog_number=536875574
   TC_idsm_chicago_prog_number=536875586
   TC_idsm_detroit_prog_number=536875587
```

Configuring multiple sites on a Windows server

You can configure Multi-Site Collaboration to support multiple sites on a single Windows server. Siemens PLM Software recommends that you read through the entire procedure before beginning.

Start with two fully installed sites of Teamcenter configured for Multi-Site Collaboration on a single server.

Set up multiple ODS daemons on a single server

In this example, the sites are called **chicago** and **detroit**.

1. Gain administrative privileges.

2. Start a Windows command prompt.

3. Copy the **%TC_ROOT%\bin\run_tc_ods.bat** file to **%TC_ROOT%bin\run_tc_ods_chicago.bat** and **%TC_ROOT%bin\run_tc_ods_detroit.bat**.

4. Edit the **%TC_ROOT%\bin\run_tc_ods_chicago.bat** file by adding this argument to the ODS run line:

   ```
   %TC_ROOT%\bin\ods.exe rpc_prog_number=536875585
   ```

5. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **chicago** site is installed.

6. Confirm that the line that calls the **tc_profilevars.bat** file points to the correct **TC_DATA** directory.

7. Edit the **%TC_ROOT%\bin\run_tc_ods_detroit.bat** file by adding this argument to the ODS run line:

   ```
   %TC_ROOT%\bin\ods.exe rpc_prog_number=536875584
   ```

8. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **detroit** site is installed.

9. Confirm that the line that calls the **tc_profilevars.bat** file points to the correct **TC_DATA** directory.

   The Windows services that control the ODS is now modified. You must have a utility that can add and delete Windows services. In the following example, the **sc.exe** program is used. This program is standard on most Windows versions and is available with the Windows Resource kit.

10. Delete the current ODS service by entering the following command:

    ```
    sc "gs_ods" delete
    ```

11. Create the ODS service for **chicago** by entering the following command:

Note

You must type the full path name, not the environment variable substitution **%TC_ROOT%**. The name of the **run_tc_ods.bat** file must match the service name. For example, **run_tc_ods_chicago.bat** and service name **gs_ods_chicago**.

```
sc create gs_ods_chicago binpath=%TC_ROOT%\bin\gs_service.exe
```

12. Create the ODS service for **detroit** by entering the following command:

    ```
    sc create gs_ods_detroit binpath=%TC_ROOT%\bin\gs_service.exe
    ```

13. Access the **Services** dialog box from the Windows control panel. Select the services you created (**gs_ods_chicago** and **gs_ods_detroit**) and change the display name to something appropriate. Fox example, ODS Service for **chicago** and ODS Service for **detroit**.

14. Start each service.

    The **chicago** and **detroit** ODS daemons are now running on the same service.

Set up multiple IDSM daemons on a single server

The sites are called **chicago** and **detroit**.

1. Gain administrative privileges.

2. Copy the**%TC_ROOT%\bin\run_tc_idsm.bat** file to **%TC_ROOT%\bin\run_tc_idsm_chicago.bat** and **%TC_ROOT%\bin\run_tc_idsm_detroit.bat**.

3. Edit the **%TC_ROOT%\bin\run_tc_idsm_chicago.bat** file by adding this argument to the IDSM run line:

   ```
   %TC_ROOT%\bin\idsm.exe pmon rpc_prog_number=536875586
   ```

4. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **chicago** site is installed.

5. Confirm that the line that calls the **tc_profilevars.bat** file points to the correct **TC_DATA** directory for the **chicago** site.

6. Edit the **%TC_ROOT%\bin\run_tc_idsm_detroit.bat** file by adding this argument to the IDSM run line:

   ```
   %TC_ROOT%\bin\idsm.exe pmon rpc_prog_number=536875587
   ```

7. If needed, change the values for **TC_ROOT** and **TC_DATA** to point to where the **detroit** site is installed.

8. Confirm that the line that calls the file points to the correct **TC_DATA** directory for the **detroit** site.

   The Windows services which control the IDSM is now modified. You must have a utility that can add and delete Windows services. In the following example, the **sc.exe** program is used. This program is standard on most Windows versions and is available with the Windows Resource kit.

9. Delete the current IDSM service by entering the following command:

```
sc "gs_idsm" delete
```

10. Create the IDSM service for Chicago by entering the following command:

> **Note**
>
> You must type the full path name, not the environment variable substitution **%TC_ROOT%**. The name of the **run_tc_idsm** batch file must match the service name. For example, **run_tc_idms_chicago.bat** and service name **gs_idsm_chicago**.

```
sc create gs_idsm_chicago binpath=%TC_ROOT%\bin\gs_service.exe
```

11. Create the IDSM service for Detroit by typing the following command:

```
sc create gs_idsm_detroit binpath=%TC_ROOT%\bin\gs_service.exe
```

> **Note**
>
> If the **TC_ROOT** for Chicago and Detroit are different, the respective correct path must be used for each service entry.

12. Access the **Services** dialog box from the Windows control panel. Select the services you created (**gs_idsm_chicago** and **gs_idsm_detroit**) and change the display name to something appropriate. For example, IDSM Service for Chicago and IDSM Service for Detroit.

13. Start each service.

    The Chicago and Detroit IDSM daemons are now running on the same service.

Configuring site preferences

Configure the site preferences on both sites using the site-specific RPC program number:

TC_*daemon-name_site-name*_**prog_number=***RPCProgramNumber*

This preference file is used by all sites accessing the remote sites.

For information about setting site preferences, see the *Preferences and Environment Variables Reference*.

The default RPC program number for:

- ODS is 536875585
- IDSM is 536875586

Additional ODS servers require unique RPC program numbers; use descending numbers beginning with the default number. Multiprocess ODS configuration requires an additional block of unused RPC program numbers for proper operation. The size of the block of numbers must be equal to the **ODS_multiprocess_max_subprocess_count** site preference value. The block of numbers must start immediately after the main ODS program number and be consecutive.

Additional IDSM servers require unique RPC program number; use ascending numbers beginning with the default number.

Note

The Noblenet Portmapper Service must be started before the IDSM and ODS service.

For example, add the following site preferences to the **Data Sharing.Multi-Site Collaboration** preference category on both sites:

TC_ods_chicago_prog_number=536875585

TC_ods_detroit_prog_number=536875584

TC_idsm_chicago_prog_number=536875586

TC_idsm_detroit_prog_number=536875587

The following example illustrates multiprocess ODS in use at the Detroit site with a **ODS_multiprocess_max_subprocess_count** value of 10. Note the block of ten unused program numbers between Chicago and Detroit.

TC_ods_chicago_prog_number=536875585

TC_ods_detroit_prog_number=536875584

TC_idsm_chicago_prog_number=536875586

TC_idsm_detroit_prog_number=536875587

Using Multi-Site Collaboration through a firewall

Multi-Site Collaboration provides two methods for communicating through firewalls. You can configure a site to communicate using the HTTP or HTTPS protocol, or you can set up to use remote procedure call (RPC) technology to communicate between client and server processes. When you configure a site in the Organization application, you designate whether it is HTTP enabled or not.

When configured to communicate using HTTP/HTTPS protocol, you configure the network to handle Multi-Site Collaboration traffic as you would any HTTP traffic. There are no additional requirements for Multi-Site Collaboration to communicate through firewalls in this configuration. If you do not designate a site as HTTP-enabled, the site uses RPC technology, and therefore, must be configured to open ports for ODS and IDSM connections. The HTTP enable configuration does not require this and eliminates the associated security risk or additional configuration required to mitigate this risk. HTTP requests are sent through the services-oriented architecture (SOA) service and use single sign-on (SSO) validation. RPC requests do not use SSO validation.

Note

The definition of a site as HTTP-enabled affects outbound requests only. For inbound requests, the services the site provides determine the communication protocol. For instance, if the protocol selected at the receiving site is RPC (**Is HTTP Enabled** box is not selected), the site can still process inbound HTTP requests if it is running the SOA service through the pool manager. It can also process inbound RPC requests provided it is running the ODS/IDSM daemons.

When configured to use RPC, a Multi-Site Collaboration client initially connects to an ODS or IDSM server, the RPC portmapper dynamically assigns a communication port that is used by the client and server processes to conduct their business.

The fact that the communication port is dynamically assigned by the portmapper requires most firewalls to open up all ports above 1023 in order for Multi-Site Collaboration, and most RPC applications, to operate. This opening up of ports creates a large hole in the firewall, creating a security risk.

Multi-Site Collaboration provides a solution to this problem by making it possible to assign specific TCP/IP ports to be used by its clients and servers. For additional information, see *Install with specific port numbers*.

Adding a proxy server

For an additional level of security, you can place an IDSM proxy server between a firewall and internal sites, isolating internal sites from direct network access by external sites. This further increases security while simplifying security setup and maintenance.

Proxy server design

The IDSM proxy server runs a daemon which creates a logical connection between an external client and an internal IDSM or ODS server. The logical connection is created dynamically when required, then terminated at the end of the client-server session.

Client IDSM requests from external sites to any internal site is channeled through the IDSM proxy host which directs the request to the appropriate internal host. Conversely, data from internal sites are channeled to the IDSM Proxy host which relays the data to the appropriate external site.

The result is that the IDSM proxy host isolates all internal sites from direct network access by external sites.

Potentially, a different set of external sites may require communication to another IDSM proxy host through another firewall to communicate to the same set of internal sites. You can also place another firewall between the internal sites and the proxy host for additional security. It is also possible for each external site to have its own firewall.

Configuring site preferences with a proxy server

In the case of ODS or IDSM proxy server configuration, the correct syntax for site-specific RPC program number preference is:

TC_*daemon-name_site-ID*_**prog_number=***RPCProgramNumber*

The proxy server does not have database access, so it has no knowledge of site names.

Proxy server system requirements

The system requirements for the proxy server differs from the IDSM server. The proxy host does not have a database; it does not even have to perform import/export operations. Thus, the disk requirements are only for loading Teamcenter and providing enough virtual memory to run the proxy server processes.

Estimate system requirements by determining the average number of simultaneous server processes expected.

Multi-Site Collaboration functions available with a proxy server

When using an IDSM proxy server, there are functions and important characteristics of a Multi-Site Collaboration network you must understand.

Functionality available to external sites

Users at all external sites can perform all IDSM-related operations with the internal sites as if the external and internal sites were directly connected, including:

- Remote import operations
- Sending objects to internal sites using EPM handlers
- Synchronization operations

Though external sites are typically importing only replicas from internal sites, use of a proxy server does not limit any IDSM-related functions.

The ability of external sites to access specific sites and objects are subject to the same Multi-Site Collaboration security controls that are in effect if the external sites were directly connected to the internal sites.

Functionality available to internal sites

Users at internal sites can perform all IDSM-related operations with the external sites as if the external and internal sites were directly connected, including:

- Synchronization operations
- Sending objects to external sites using EPM handlers
- Remote import operations

Though internal sites will typically be communicating with external sites to synchronize, use of a proxy server does not limit any IDSM-related functions.

The ability of internal sites to access specific sites and objects are subject to the same Multi-Site Collaboration security controls that are in effect if the internal sites were directly connected to the external sites.

Note

The IDSM user must have write access to a master item at the owning site to make changes to remote item replicas. You must make the IDSM user a member of the dba group or change the rule tree to grant Write access. The replica revision fails with the error: **No Write access to master item**.

Requesting automatic synchronization and notification

A user performing remote import operations can request automatic synchronization and notification.

Install with specific port numbers

This section provides the instructions on how to configure ODS and IDSM to assign specific TCP/IP ports for Multi-Site Collaboration clients and servers.

Three port numbers are used by Multi-Site Collaboration when using the firewall configuration, which uses specific port number selection:

- The portmapper always uses port 111.
- The ODS uses one port for RPC commands.
- The IDSM uses one port for RPC commands.

The ODS and IDSM ports are user selectable.

Configure the ODS

The ODS server has the **-tcp_port_number=** parameter that is used to select the fixed TCP port. The selected port number must be between 1024 - 49150 and must not conflict with any existing services. Contact your system administrator if in doubt.

UNIX:

1. Edit the *TC_ROOT***/bin/run_tc_ods** script file.

   Note

   These edits must be made on both proxy server and client hosts.

2. Locate the following line:

```
nohup ${TC_ROOT}/bin/ods >
   ${TC_TMP_DIR}/ods$$.log 2>&1
```

3. Change it to:

```
nohup ${TC_ROOT}/bin/ods
   -tcp_port_number={Selected ODS Port} >
   ${TC_TMP_DIR}/ods$$.log 2>&1
```

4. Save the file.

   The changes take effect the next time you reboot.

Windows:

1. Edit the **%TC_BIN%\run_tc_ods.bat** script file.
2. Locate the following line:

```
TC_ROOT\bin\ods.exe
```

3. Change it to :

```
TC_ROOT\bin\ods.exe -tcp_port_number={Selected ODS Port}
```

4. Save the file.

   The changes take effect the next time you reboot.

Configure the IDSM

The IDSM also has the **-tcp_port_number** parameter described in the *Configure the ODS* section. The **idsminetd** UNIX utility performs the task of launching the IDSM server from the standard UNIX **inetd** utility.

Note

The port number must be different from the ODS port number.

idsminetd utility

On UNIX systems, the **idsminetd** utility serves as the IDSM launching program. Located in **$TC_ROOT/bin** directory, it is run at system startup and services all inbound requests for a new IDSM.

The **idsminetd** utility has the following command format:

idsminetd [-dt] [-p=*tcp_port_number***]**
[-r=*IDSM start script***]**

Option	Descriptions
-d	Debug mode for standalone testing.
-t	Enhanced logging.
-p	Specify the port number the IDSM should run on. The default is the system-assigned port number.
-n	Specify the RPC program number the IDSM should use. The default RPC program number is used if this argument is omitted. If you use this argument, you must set the **TC**_*daemon-name_site-name*_**prog_number** site preference to the value you specify. For example, if you specify –**n=536875586** in the **idsminetd** command, set the preference as: `TC_idsm_chicago_prog_number=60003` For more information about the **idsminetd** utility, see the *Utilities Reference*.
-r	Specify the IDSM start script. The default is *TC_ROOT***bin/run_tc_idsm**.

In normal mode, all output is sent to **syslogd**. In debug mode, output is sent to **stderr**.

UNIX:

1. Create a **rc.ugs.idsminetd** script based on the template in sample **rc.ugs.idsminetd** script.
2. Modify **TC_ROOT** and **TC_DATA** to match your configuration. These entries appear twice.
3. Modify the following line to use the selected IDSM port number:

```
nohup ${TC_ROOT}/bin/idsminetd
   -p=Selected IDSM Port
```

```
-r=${TC_ROOT}/bin/run_tc_idsm > /tmp/idsminetd$$.log &
```

4. Enter the following command to make the script executable:

```
chmod 555 rc.ugs.idsminetd
```

5. Based on the platform, copy the **rc.ugs.idsminetd** script to the following platform and system directory:

Platform	Directory	Notes
IBM-AIX	**/etc/rc.ugs.idsminetd**	Modify the **/etc/rc.ugs** script to run the **/etc/rc.ugs.IDSM** script at system startup.
HP-UX	**/sbin/init.d/rc.ugs.idsminetd**	
SUN Solaris	**/etc/init.d/rc.ugs.idsminetd**	
Linux	**/etc/init.d//rc.ugs.idsminetd**	

6. Create a numbered symbolic link to **rc.ugs.idsminetd** in the appropriate run control directory:

Platform	Directory	Notes
IBM-AIX	Not needed on AIX	Modify the **/etc/rc.ugs** script to run the **/etc/rc.ugs.IDSM** script at system startup.
SUN Solaris	**/etc/rc3.d/S99rc.ugs.idsminetd**	
HP-UX	**/sbin/rc3.d/S909rc.ugs.idsminetd**	This starts the **idsminetd** process owned by administrator user and listens for IDSM requests.
Linux	**/etc/init.d/rc5.d/S21rc.ugs.idsminetd**	

Windows:

1. Edit the **%TC_BIN%\run_tc_idsm.bat** script file.

   Note

   These edits must be made on both proxy server and client hosts.

2. Locate the following line:

```
TC_ROOT\bin\idsm.exe pmon
```

3. Change it to:

```
TC_ROOT\bin\idsm.exe pmon
   -tcp_port_number=Selected IDSM Port
```

4. Save the file. The changes take effect the next time you reboot.

Configuring an ODS and IDSM proxy server

Because the proxy server does not have a database, it does not define sites in the same manner as other Multi-Site Collaboration servers. Instead, the site information is stored in an XML file.

When a requesting site sends an RPC message to the target site through the proxy host, using the **version_check_RPC** function, only the target's site ID is delivered in the RPC message, not the site name. A preference is required to map the target's site ID with its actual node name so that the proxy server can redirect the message to the actual node. The same is true of the ODS setup.

Configure proxy servers

You install the Multi-Site Collaboration Proxy functionality using the Teamcenter Environment Manager (TEM) installer.

1. Launch the TEM installer.
   a. Choose to create a new installation of the product.
   b. Select the Multi-Site Collaboration Proxy Server Solution and complete the configuration with the TEM installer.
   c. Type the target directory for **TC_ROOT** in the installation directory box.

      Caution

      Do not select any other solution with the Proxy Server Solution.

   d. Type the password for the operating system user ID used to launch these daemons/services.
2. Define the ODS and IDSM proxy server site tables and proxy server types.

   Note

   Because a proxy server does not use a database, it uses the **tc_preferences_overlay.xml** file to store Teamcenter site preferences. You must manually edit this file to set the preference values.

   Modify the **TC_ods_proxy_server_site_table** and **TC_idsm_proxy_server_site_table** entries defined in the **tc_preferences_overlay.xml** file located in the *TC_DATA* directory.

   Note

   The *TC_DATA* directory is not explicitly defined in TEM when you install a proxy server. TEM creates this directory under the *TC_ROOT* directory and names the directory using the value you specify in the **ID** box in the **New Configuration** panel.

   Set the **IDSM_proxy_server_type** and **ODS_proxy_server_type** values to **Relay**.

   These settings must include all internal and external sites that will use the proxy host. Generally this means there must be at least one entry for

each site in a Multi-Site Collaboration federation. Sites not noted in this preference cannot use this host as a proxy.

The format for the value of this preference is:

*site-id1***:***real-node-name-for-site-id1*
*site-id2***:***real-node-name-for-site-id2*

or

*site-id1***:***IP-address-for-site-id1*
*site-id2***:***IP-address-for-site-id2*

site-id1 is the site ID of an internal or external site which uses the proxy host.

real-node-name-for-site-id2 is the actual node name for the site ID. An IP address can be given instead of a node name.

The colon is a separator between the site ID and the node name.

The following are sample XML preferences for this four-site setup:

Site A:

Site id: 183853823

Hostname: mainnode1

Site B:

Site id: 210103239

IP address:134.244.96.171

Site C:

Site id: 174090661

Hostname:suppliernode1

Site D:

Site id: 354153256

IP address:144.132.44.153

```
<preference name="TC_idsm_proxy_server_site_table"
 type="String" array="true" disabled="false">
<preference_description>Specifies the list of sites
   that are allowed to access an IDSM proxy server. This is valid
   only at the Proxy server node and only when IDSM_proxy_server_type=Relay.
   Format is <Site ID>:<Node Name>. Example: 123456789:sun_node1 or
   123456789:111.222.33.444.</preference_description>
<context name="Teamcenter">
```

<value>183853823:mainnode1</value>

<value>210103239:134.244.96.171</value>

<value>174090661:suppliernode1</value>

<value>354153256:144.132.44.153</value>

```
</context>

</preference>

<preference name="TC_ods_proxy_server_site_table" type="String"
   array="true" disabled="false">
<preference_description>Specifies the list of sites that are
   allowed to access an ODS proxy server. This is valid only at the
   Proxy server node and only when ODS_proxy_server_type=Relay.
   Format is <Site ID>:<Node Name>. Example: 123456789:sun_node1 or
   123456789:111.222.33.444.. Example: 123456789:sun_node1 or
   123456789:111.222.33.444.</preference_description>
```

```
<context name="Teamcenter">
<value>183853823:mainnode1</value>
<value>210103239:134.244.96.171</value>
<value>174090661:suppliernode1</value>
<value>354153256:144.132.44.153</value>
</context>
</preference>
```

Configure a proxy client

After the ODS, the IDSM and the proxy server are configured, the proxy clients can be configured. The internal and external sites are considered clients of the proxy host.

You must define each site which is part of the Multi-Site Collaboration federation at each local database. Each site must be assigned a site name and a host name, also called a *node name*. A site name can contain up to 128 characters.

Naming of clients differ depending on whether a proxy server is being used as explained in the following table:

Firewall configuration	Requirements
Without a proxy server	Each external site must define each internal site with their respective host names.
With a proxy server	Each external site must define any internal site with the host name of the proxy server. Each internal site must define any external site with the host name of the proxy server.

1. Set up the site as a Multi-Site Collaboration working site.
2. For internal sites, define the external sites that are allowed access to the internal site:
   - Start the Organization application and select **Sites** from the list in the **Organization** tree pane.
   - In the **Sites** pane, type the desired values in the **Site Name** and **Site ID** boxes. A site name can contain up to 128 characters.
   - Type the name of the proxy host in the **Site Node/URL** box.
3. For external sites, define the internal sites that are accessed using the proxy host:
   - From the Organization application, select **Sites** from the list in the **Organization** tree pane.
   - Enter the **Site Name** and **Site ID** as you would normally for a regular remote site. A site name can contain up to 128 characters.
   - Enter the name of the proxy host in the **Site Node/URL** text field.

4. For each site, define the following IDSM preferences to control access by a remote site:

   IDSM_permitted_sites
   IDSM_permitted_transfer_sites

   Note

   The proxy host is not a site itself; it should not be included in these site preferences.

5. (Optional) For each client site, set an alternate proxy host by defining the following site preference:

   IDSM_proxy_client_alternate_proxy_host_for_
   *Proxy-host-node-name***=**
   alternate-Proxy-host node-name

   For example, if the proxy hosts node name is **myproxy1** and the node name of the alternate is **myproxy2**, the site preference would be defined as:

   IDSM_proxy_client_alternate_proxy_host_for_
   myproxy1=myproxy2

   Note

   Siemens PLM Software recommends that all proxy-related preferences at client sites are prefixed by **IDSM_proxy_client** while those required at the proxy host are prefixed by **IDSM_proxy_server**.

Bypass portmapper service

This service allows you to bypass the portmapper service, **rpcbind**, which runs on port 111. You may want to bypass the portmapper service for security reasons.

Note

Before configuring an IDSM or ODS for a site using the default portmapper bypass setup, use the **rpcinfo** system utility to verify that there is no existing IDSM RPC configuration on the host. Any existing configuration must be removed to avoid conflicts with the default bypass portmapper configuration.

Bypass the portmapper service by enabling the **TC**_*daemon name_site name*_**port_number** site preference. The site preference must be set for each site to specify the port number used by clients to contact the remote ODS and IDSM servers. For additional information regarding this preference, see the *Preferences and Environment Variables Reference*.

Note

Bypassing the RPC service requires firewall configurations for both the ODS and IDSM servers.

IDSM launching utility

On UNIX systems, the **idsminetd** utility serves as the IDSM launching program. Located in the **$TC_ROOT/bin** directory, it is run at system startup and services all inbound requests for a new IDSM.

The **idsminetd** utility has the following command format:

idsminetd [-dt] [-p=tcp_port_number] [-r=idsm start script]

Option	Description
-d	Debug mode for standalone testing.
-t	Enhanced logging.
-p	Specify the port number the IDSM should run on. The default is the system-assigned port number.
-n	Specify the RPC program number the IDSM should use. The default RPC program number is used if this argument is omitted.
-r	Specify the IDSM start script. The default is *TC_ROOT*/**bin/run_tc_idsm**.

In normal mode, all output is sent to the **syslogd**. In debug mode, output is sent to **stderr**.

UNIX configuration

1. Create a **rc.ugs.idsminetd** script based on the template in **rc.ugs.idsminetd**.

2. Modify **TC_ROOT** and **TC_DATA** to match your configuration. These entries appear twice.

3. Modify the following line to use the selected IDSM port number:

```
nohup ${TC_ROOT}/bin/idsminetd
   -p=Selected IDSM Port
   -r=${TC_ROOT}/bin/run_tc_idsm > /tmp/idsminetd$$.log &
```

4. Enter the following command to make the script executable:

```
chmod 555 rc.ugs.idsminetd
```

5. Type the following command to set the IDSM rpc program number using the **-n** argument:

```
nohup ${TC_ROOT}/bin/idsminetd
-p=selected-IDSM-port
-n=IDSM-rpc-program-number
-r=${TC_ROOT}/bin/run_tc_idsm > /tmp/idsminetd$$.log &
```

6. Based on the platform, copy the **rc.ugs.idsminetd** script to the following platform and system directory:

Platform	Directory	Notes
IBM	**/etc/rc.ugs.idsminetd**	Modify the **/etc/rc.ugs** script to run the **/etc/rc.ugs.idsm** script at system startup.
HP-UX	**/sbin/init.d/rc.ugs.idsminetd**	
SUN Solaris	**/etc/init.d/rc.ugs.idsminetd**	
Linux	**/etc/init.d/rc.ugs.idsminetd**	

7. Create a numbered symbolic link to **rc.ugs.idsminetd** in the appropriate run control directory:

Platform	Directory	Notes
IBM-AIX		Not needed on AIX
HP-UX	**/sbin/rc3.d/S909rc.ugs.idsminetd**	
SUN Solaris	**/etc/rc3.d/S99rc.ugs.idsminetd**	
Linux	**/etc/init.d/rc5.d/S21rc.ugs.idsminetd**	

8. Verify that the IDSM and ODS are running after a system restart and manually start them if required.

Windows configuration

1. Edit the **%TC_BIN%\run_tc_idsm.bat** script file.
2. Locate the following line:

   ```
   TC_ROOT\bin\idsm.exe pmon
   ```

3. Change the line to:

   ```
   TC_ROOT\bin\idsm.exe pmon -tcp_port_number={Selected IDSM Port}
   ```

4. Save the file. The changes take effect the next time you reboot.

Sample rc.ugs.idsminetd script

```
#!/bin/ksh
# Set default return status - 0 for success.
rval=0
#
[ -x $0 ] || {
# Execute permission check, if not, quietly exit.
echo "\nERROR:\tNo permission to execute $0."
rval=2
exit $rval
}
#
# Set PATH environment variable
PATH=/usr/sbin:/usr/bin:/sbin:/bin:/etc ; export PATH
#
```

```
# Set TC_ROOT environment variable.
TC_ROOT=Your TC_ROOT path; export TC_ROOT
# Set TC_DATA environment variable.
TC_DATA=Your TC_DATA path; export TC_DATA
#
# Check if the daemon is there and executable.
if [ -x $TC_ROOT/bin/idsminetd ]
then
echo " Starting idsminetd daemon on `uname -n`"
su infodba -c /bin/ksh <<-EOF
TC_ROOT=Your TC_ROOT path
export TC_ROOT
TC_DATA=Your TC_DATA path
export TC_DATA
TC_TMP_DIR=/tmp
export TC_TMP_DIR
. ${TC_DATA}/tc_profilevars
nohup ${TC_ROOT}/bin/idsminetd -p=Selected_IDSM_PORT -
r=/etc/run_tc_idsm > /tmp/idsminetd$$.log &
sleep 8
EOF
else
# idsminetd daemon is missing - error
echo "\nERROR:\tUnable to locate idsminetd daemon."
rval=1
fi
# Check to see if daemon is still running. If not, fail.
if [ `ps -ef|grep idsminetd|grep -v grep|wc -l` -eq 0 ] ; then
echo "\nERROR:\tFailed to start idsminetd daemon."
echo "\tPlease refer to the syslog for error messages."
rval=1
fi
exit $rval
```

Configure HTTP enabled Multi-Site

When configured to communicate using HTTP/HTTPS protocol, configure the network to handle Multi-Site Collaboration traffic as you would any HTTP traffic. If you do not designate a site as HTTP enabled, the site uses RPC technology, and therefore, must be configured to open ports for ODS and IDSM connections. The HTTP enable configuration does not require this and eliminates the associated security risk or additional configuration required to mitigate this risk. To create an HTTP enabled site:

1. Select the top-level sites node  from the **Organization List** tree.

2. In the **Sites** pane, type a descriptive name for the site in the **Site Name** box. For example, Teamcenter site 1 defines two remote sites: **TcHost2** and **TcEntHost**.

3. Type a unique identifier in the **Site ID** box. For example, type **457655709** for the **TcHost2** site.

   Caution

   Each site must be defined at other sites using exactly the same site ID in each definition.

4. Type the URL used to contact the thin client (Web application) solution for the site in the **Site Node/URL** box. For example, if you deployed the Web application that connects to the **TcHost2** site on a WebLogic application server on that host

using default values for the application name and the application server listener port, type **http://TcHost2:7001/tc**.

5. (Optional) Select the **Provide Object Directory Services** check box if this is an ODS site.

6. (Optional) Select the **Is A Hub** check box if this is a hub site.

7. Select the **HTTP Enabled Multisite** check box.

8. (Optional) Select the **Allow deletion of replicated master object to this site** check box. Selecting this allows deleting master objects which have been replicated to the site, even if there is a replica existing for this master object at the site.

9. Click **Create**.

The site is created and appears in the **Organization List** tree.

Configure Multi-Site for HTTPS

You can enable Multi-Site Collaboration support for HTTPS communication by setting the **TEAMCENTER_SSL_CERT_FILE** environment variable to the location of an SSL certificate authority (CA) file. This file must contain one or more certificates in Privacy-Enhanced Mail (PEM) format.

The CA file must contain one or more certificates as follows:

```
-----BEGIN CERTIFICATE-----
  ⋮
   (CA certificate in base64 encoding) ...
  ⋮
-----END CERTIFICATE-----
```

You can provide text before, after, or within the certificate that provides information about the certificate, for example:

```
-----BEGIN CERTIFICATE-----

   A PEM (.pem) format digital certificate, base 64 encoding:
   MB4CGQDUoLoCULb9LsYm5+/WN992xxbiLQlEuIsCAQM=
-----END CERTIFICATE-----
```

To configure the **data_share** process for an HTTPS-based Multi-Site Collaboration environment:

1. Generate a security certificate (CRT) file.

   a. At a command prompt, type:

      ```
      keytool -keystore /usr/java/j2re1.4.2_07/lib/security/cacerts
      -export -alias verisignserverca > /tmp/verisign.cacert
      ```

   b. View the binary certificate (DER) file to verify that is was created, type:

      ```
      openssl x509 -noout -text -in /tmp/verisign.cacert -inform der
      ```

   c. Convert the DER file to PEM format, type:

      ```
      openssl x509 -out /tmp/verisign-cacert.pem
      -outform pem -text -in /tmp/verisign.cacert -inform der
      ```

2. Open the **tc_profilevars** file and locate the **TC_DATA** variable setting.

3. Immediately following the **TC_DATA** variable, set the **TEAMCENTER_SSL_CERT_FILE** environment variable as follows:

   ```
   TEAMCENTER_SSL_CERT_FILE=${TC_DATA}/pom_transmit/ca-bundle.crt;
   export TEAMCENTER_SSL_CERT_FILE
   ```

   In this example, **ca-bundle.crt** is the CA file located under the **pom_transmit** directory.

For additional information about enabling SSL for Teamcenter applications, see *Security Services Installation/Customization*.

Configure HTTP enabled Multi-Site for single sign-on

If you use Security Services single sign-on (SSO) functionality, configure the **TC_SSO_app_id_of_site**_*site-name* preference at the site. *site-name* represents the name of the site and this preference does not exist by default. The preference value must match the **Application ID** value for the site as defined in the Application Registry table.

For information about the Application Registry table, see *Security Services Installation/Customization*. For information about creating and setting preferences, see the *Preferences and Environment Variables Reference*.

Note

All sites that are using SSO must be in the same SSO domain.

Using HTTP enabled Multi-Site with forward proxy servers

Your Teamcenter thin client can use the browser proxy support, Java plug-in integration to support a forward proxy server when using HTTP enable Multi-Site. For rich client configuration, you can use one of the following approaches:

- Use Teamcenter Security Services forward proxy support.
- Configure the proxy related Java properties for the clients.

For basic HTTP forward proxy authentication, you must use Teamcenter Security Services. File Management System (FMS) does not support authentication on server integrations. Therefore, you must use the RPC proxy server configuration for authenticated server communications in your Multi-Site environment as described in *Adding a proxy server*.

For information about configuring Security Services, see *Security Services Installation/Customization*.

Note

Multi-Site uses FMS for file transfers. Therefore, you must configure Multi-Site for FMS whether you use a proxy server or not.

For more information, see *Configure FMS*.

Customizing an ODS schema

You can add custom attributes to the publication record for an ODS to allow sites with differing schemas to publish and search for shared data. This also allows a different type of PLM site, such as Teamcenter Enterprise, to participate in data sharing as an ODS client. A custom attribute can be declared as mandatory or optional by the client.

Custom attributes for a publication requires the following:

- A customization of the client and server side to support custom attributes.
- The ODS servers publication record schema must be a superset of all client schemas.
- The POM name and type of all custom attributes must match the attributes in the ODS server.
- All custom attributes must be of type string if you use the site preferences to identify the attributes published to a server. This limitation is removed when using user exits to process custom attributes.
- A separate saved query for each of the different ODS publication record schemas.
- The ODS server must support any mandatory custom attributes to prevent the failure of an operation that includes the attribute.

  Note

  A query including mandatory custom attributes fails only on the ODS servers that do not support the attribute. An error is returned from that ODS server only. The query returns results, as expected, for ODS servers that support the attribute.

There are two site preferences associated with custom string attributes for a publication record. The **TC_ods_client_extra_attributes** value indicates which attributes at the client side are published to the server. The **PublishedObjConfiguredProperties** global constant value contains name pairs that indicates the display names of custom attributes.

For information about setting preferences, see the *Preferences and Environment Variables Reference*.

You can use the following preferences to control the display of custom attributes in My Teamcenter:

- **PUBLISHEDOBJECT_object_columns_hidden**

  Specifies the list of column names that are hidden for **PublishedObject** objects. The following are hidden by default:

  po_object_class
  po_owner_id
  po_group_id
  po_object_creation_date
  po_pub_date

- **PUBLISHEDOBJECT_object_columns_shown**

Specifies the list of column names that are shown for **PublishedObject** objects. The following are shown by default:

po_object_id
po_object_rev_id
po_object_name
po_object_type
po_owning_site
po_object_rel_stat_names
po_object_desc

- **PUBLISHEDOBJECT_object_widths_hidden**

  Specifies the width of column names that are hidden for **PublishedObject** objects. The default width for the default hidden columns is **32**.

- **PUBLISHEDOBJECT_object_widths_shown**

  Specifies the width of column names that are shown for **PublishedObject** objects. The default width for the default shown columns is **32**.

Add custom string attributes

This procedure uses the Business Modeler IDE to update the publication record by adding all custom attributes to the ODS server and custom attributes supported by the ODS client site to its publication record schema.

For information about adding attributes to a class, see the *Business Modeler IDE Guide*.

1. Access the **Business Modeler IDE** perspective by choosing **Window→Open Perspective→Other→Business Modeler IDE**.

2. Choose **File→New→Project** and use the New Project wizard to create a project for your customization.

3. Click the **Classes** tab to display the **Classes** view.

4. Browse to the **PublicationRecord** class or click the **Find Class** button and search for the class.

5. Right-click the **PublicationRecord** class and choose **Open**. A view displays the class details and attributes.

6. To add an attribute, click the **Add** button to the right of the **Attributes** table.

   Perform the following steps in the **Class Attribute** dialog box:

   a. In the **Name** box, type the attribute name.

      When you name a new data model object, you should add a prefix to the name to designate the object as belonging to your organization, such as a three-letter acronym.

   b. In the **Attribute Type** box, select the **String** storage type.

   c. In the **String Size** box, type the character length of the attribute.

d. In the **Keys** area, check the **Nulls Allowed** property.

e. Click **Finish**.

The new attribute appears in the properties table and are marked with a **c** indicating it is a custom attribute.

f. For the ODS server, continue to add custom string attributes in this manner until you have added all custom attributes from all client sites.

Note

If you add publication record class attributes to the local site publication record with the same attribute names as used in the server, you can then use the query builder on the client side to build a saved query that can be executed directly by the server.

Add custom nonstring attributes

You implement user exits to add nonstring custom attributes or attributes that are not part of the object POM attribute list or master form. User exits can be used in addition to custom attributes added through the **TC_ods_client_extra_attributes** preference. To support these type of custom attributes, use the Integration Toolkit (ITK) to implement custom behavior for the following user exits:

User_ods_client_ask_extra_attribute_names

This user exit is called prior to registering the client schema with the server. It adds the list of attribute names to the list of attribute names from the **TC_ods_client_extra_attributes** preference, if any, and then sends it to the server as part of the schema registration process.

User_ods_client_publish_extra_attributes

This user exit is called prior sending the publication request to the ODS server. The implementation must convert nonstring attribute values to string values and the attribute names returned must match:

- The attribute names returned by the **User_ods_client_ask_extra_attribute_names** user exit.
- The attribute names added to the local publication record.
- The attribute names added to the ODS server publication record.

For information about customization and user exits, see the *Server Customization Programmer's Guide*.

Customizing dataset export behavior

Multi-Site Collaboration supports controlling whether to export a dataset or not through relationship attachments. You can also use the **USER_is_dataset_exportable** user exit to control export decision for datasets. For example, consider the following use case:

Item1
—ItemRevision1

— **Dataset1** (attached to **ItemRevision1** with **References** relation)
— **Dataset2** (attached to **ItemRevision1** with **References** relation)

A user attempting to export the **Item1** object to remote site can select the relationships to include in the export. Selecting the **Reference** relationship causes Multi-Site to export both datasets (**Dataset1** and **Dataset2**) with the object. Using relationships attachments as the filtering mechanism, it is not possible to export one dataset without the other.

This **USER_is_dataset_exportable** user exit provides a secondary level of filtering in a Multi-Site export transaction. Multi-Site invokes this user exit during each export transaction for each related dataset. The user exit contains your custom code that receives the dataset tag, target site information, and other relevant information about transaction. You can implement the appropriate business logic that provides a **true** (yes export) or **false** (no do not export) value in the **isExportable** output parameter. Multi-Site determines whether to include the dataset in the export depending on the parameter value returned from the user exit.

Note

- The user exit is invoked for each dataset version being exported to remote site. Your custom code must be consistent about yes/no decisions for all versions.

- The exporting site is responsible for making export/no-export decisions. Therefore, the user exit is always invoked at the exporting site and the importing site is not involved in the decision process.

- Multi-Site designates certain relationships as required. The **USER_is_dataset_exportable** user exit is not invoked for required relationships.

  If the user exit does not return **ITK_ok**, the export/no-export decision from the user exit is ignored.

. .

For information about customization and user exits, see the *Server Customization Programmer's Guide*.

Chapter

9 Troubleshooting reference

Chapter

9 Troubleshooting reference

This information is intended to help anyone that is involved in supporting Multi-Site Collaboration troubleshoot a problem. It also contains information to help resolve import/export problems encountered while using the import/export commands from the rich client, and the **item_export** and **item_import** command line utilities. For detailed information, see *Common import/export problems*.

Although most of the examples presented in this reference are for UNIX, such as, path names and environment variables, this information also applies to Windows systems. For Windows-specific information, see *Windows platform notes*.

The examples use 1 ODS site and 2 IDSM sites:

- ODS1 uses **node1** to run the ODS daemon with Site ID 111111111; the database server is **dbnode1**.
- Site2 uses **node2** to run the IDSM daemon with Site ID 222222222; the database server is **dbnode2**.
- Site3 uses **node3** to run the IDSM daemon with Site ID 333333333; the database server is **dbnode3**.

The following information applies to each IDSM site used in the examples:

- The transfer area is the directory defined by the **TC_transfer_area** site preference and is assumed to be the **/users/tc_transfer_area** directory.
- The IDSM server processes run in the context of the administrator user, the default value in the **inetd.conf** file is discussed in *Unable to connect to an IDSM server error*.
- The IDSM sites allow each other to transfer ownership of objects between themselves.
- Each IDSM site allows the publication of items, datasets, and forms.

Before reading about any specific problems, see *Generating complete log files* and *Interpreting the error stack* to obtain the basic background information about debugging techniques.

Error recovery

For information about rules and restrictions regarding object replication, see *Data replication*.

These rules address potential problems that can result from the uncontrolled use of replication and lack of network-wide referential integrity. However, there are situations when it is necessary to circumvent these rules to correct a more serious problem. Therefore, system administrators can now fix certain problems.

Warning

With the exception of checkpoint transactions, the information provided in the following sections is intended to help you solve occasional problems. These techniques should never be incorporated into routine site maintenance.

Using checkpoints

The **data_share** and **data_sync** utilities support checkpoint transactions that can be restarted at a failure point. Siemens PLM Software recommends that you use checkpoints if your transaction has multiple target sites. The following are valid checkpoint arguments:

Note

Entries in parentheses are accepted abbreviations for arguments.

-checkpoint (cp)

Initiates a checkpoint transaction, that is, a transaction that can be restarted at the point of failing.

It is valid only with **send** function. It is not valid with the **-transfer** argument.

If a noncheckpoint operation is initiated for multiple target sites and some target sites are not currently available based on a preliminary availability check, Teamcenter sends a message to the **stdout** device to notify the user about unavailable sites, removes unavailable sites from the target site list, and then performs the operation for the available sites.

-cleanup_transaction (ct)

Deletes transient data generated during a checkpoint transaction. Transient data consists of the export data and supporting directories and files used to manage the transaction.

-commit_ixr (cmi)

Creates or updates import export records (IXRs) in the owning site's Teamcenter database for objects in the transaction that failed. Use this only after you are sure the transaction has completed successfully.

-list_transactions (lt)

Returns a list of all transactions that have transient data that has not been deleted from the node's transfer area where this command is executed.

-status (stat)

Displays the status of a transaction. Requires the **-transaction_id** argument.

-compress_ind_files (cif)

Specifies the compression mode used for a checkpoint transaction. Valid values are, **S** (single zip file), **I** (individual zip files), and **N** (no compression).

-restart (rs)

Restarts the transaction at the point of failure. Valid only with the **-f=send** function.

-transaction_id (trid)

Specifies the 14-character transaction ID for a specific checkpoint-related operation. Use the **-list_transactions** argument to determine the transaction IDs.

-transaction_id (trid)

Specifies the 14-character transaction ID for a specific checkpoint-related operation. Use the **-list_transactions** argument to determine the transaction IDs.

For information about the using utilities, see the *Utilities Reference*.

Export recovery

When performing a remote import with transfer of ownership, Multi-Site Collaboration first exports the object from the remote site into a metafile, then copies the metafile over the network in preparation for import into the local site.

If an error occurs after the export but before the object is imported, the object is in a state where it is not owned by any site. Multi-Site Collaboration attempts to automatically recover from such an error by restoring ownership at the remote site using the metafile located at the remote site's transfer area. However, there could be cases when automatic recovery is not feasible, in which case a manual recovery must be used.

The requirement to perform a manual recovery is indicated by an error message such as:

```
Objects exported from site Design Center but not imported
at site Manufacturing Center.
Files left at site Design Center in /tmp/tc_1339_34356012
on node_hp1.
Use this directory to import at destination site or use
export_recovery at original site.
```

In the previous example, the error occurred while importing an object with ownership transfer from Design Center to Manufacturing Center. If the message states that the files were left at Manufacturing Center, which is the destination, then you must import the named directory at Manufacturing Center. Choose **Tools→Import→Objects**. If the files were left at Design Center and the transfer was initiated online, then you must run the **ensure_site_consistency** utility at Design Center on node **node_hp1**. If the transfer was initiated offline, you must run the **export_recovery** utility at the Design Center.

If data compression is enabled, you can compress the files in the **Export** directory. The **.zip** file extension indicates the files are compressed. Before recovering the data, you must first use the **decompress.pl PERL** script in the **TC_BIN** directory. The command format is:

```
perl decompress.pl export directory name
```

Recover a lost or corrupted master object

When the master object is corrupted or lost, you can recover it by exporting a replica from a remote site and importing it into the last known owning site. However, because there is a restriction about exporting replicas, a special procedure is required to make this possible.

> **Warning**
>
> This procedure is intended to be a last resort of error recovery. It should not be used for any other reason as it could lead to other equally serious problems such as multiple master copies or overwriting the latest copy with an obsolete copy. This procedure must be performed by a user with administrator user privileges, (**infodba** by default). No other user accounts have all the required privileges.

1. Set the **TC_EXPORT_COPY** environment variable to any value.
2. Export a replica of the lost object from a remote site to the last known owning site using the **Tools® Export® Objects** option or any Integration Toolkit (ITK) program that performs export. Ensure you transfer site ownership to the last known owning site.
3. Import this exported replica into the last known owning site.

Delete a master object

Under normal conditions, a master object cannot be deleted once it has been replicated to other sites in order to preserve network-wide referential integrity. However, there are times when a master object must be deleted.

1. Delete all known replicas.
2. At the owning site, run the **data_sync** utility with the **-verify** and **-update** arguments.

   This deletes any export record associated with the master object. To avoid synchronization errors, delete attached datasets first.
3. Manually delete the master object.

Recovering data due to failed transfer of ownership

In cases where legitimate error conditions are encountered during an ownership transfer (such as lack of transfer privilege or duplicate item IDs), there is normally no need to perform any corrective action; Multi-Site restores the data to consistent states under most noncrash conditions. The owning site for an object can be corrupted when a site ownership transaction that uses the Synchronous Site Transfer (SST) protocol is interrupted due to a system/network crash or a user-initiated process termination (such as by the Windows Task Manager). To correct the ownership inconsistency, use the **ensure_site_consistency** utility to perform corrective actions.

The **ensure_site_consistency** utility is context-sensitive using context information stored in an SST recovery dataset. This dataset is attached to the root or main object of a site transfer operation through the **TC_sst_record** GRM relation type. The context information is stored in the description of this dataset. Because the dataset is owned by the administrator user and cannot be deleted by a regular user, the GRM relation is *secured*.

Caution

Do not modify or delete the SST recovery dataset. This prevents the **ensure_site_consistency** utility from taking corrective action on the primary object of the **TC_sst_record** relation. The **ensure_site_consistency** utility deletes this dataset after completing the recovery.

Use the **ensure_site_consistency** utility on items that are flagged as requiring corrective action; *never* use it on an item that is not flagged.

Use the **ensure_site_consistency** utility at the exporting site only, *never* at the importing site. The *flag* that marks an object as requiring corrective action is always at the exporting site.

Determine objects requiring corrective action

There are two ways to find objects that flagged for corrective action:

- Use the **__Objects_for_Site_Consistency** saved query in a Teamcenter rich client. You must have Teamcenter administrator privileges to use this query.
- Use the **report** function of the **ensure_site_consistency** utility, for example:

```
ensure_site_consistency -f=report -search -report=recovery_candidates.txt
```

For information about the **ensure_site_consistency** utility and its functions and arguments, see the *Utilities Reference*.

After you have the list of objects, use the **ensure_site_consistency** utility to correct the problem.

- To perform corrective action on a single, type:

```
ensure_site_consistency -f=recovery -item_id=Item123
```

- To perform corrective action on all objects that require corrective actions and report the results, type:

```
ensure_site_consistency -f=recovery -search
-report=recovered-objects-list.txt
```

Note

This is the recommended best practice especially in the case of failed Remote Checkin of multiple objects.

- To perform corrective action on a list of items, type:

```
ensure_site_consistency -f=recovery -filename=item-id-list.txt
```

item-id-list.txt represents the name of a text file that contains a list of item IDs with one item ID per line.

- To perform corrective action on all objects in a uniquely-named folder, type:

  ```
  ensure_site_consistency -f=recovery -folder=unique-folder-name
  ```

  Note

  The use of a folder is suitable for workspace objects that do not have unique IDs such datasets and forms. This is useful for failed remote checkins of multiple objects where many of the remotely checked out objects do not have unique IDs (such as datasets, forms, BVRs, and so forth).

  Caution

  If a site ownership transfer is interrupted due to process termination (hardware crash or process crash), there is some lag time before Oracle detects the termination of the client process. For example, if the **data_share** process performing an export is terminated by the user pressing Ctrl+C, running the **ensure_site_consistency** utility may return a message that the recovery cannot be performed because the site ownership transaction is still in progress.

  If this happens, wait for about 20 to 30 minutes before running the **ensure_site_consistency** utility again. The length of the wait depends on the Oracle **SQLNET.EXPIRE_TIME** setting that represents the interval Oracle uses when verifying whether logged-on client processes are still active. This is normally set to 10 to 15 minutes but it typically take twice the default value before Oracle detects the termination of a client process.

  If the problem persists for a period significantly beyond the default setting, report the problem to your database administrator.

For an offline transfer of ownership, the SST protocol is not used. Offline transfer of ownerships are started using either the **item_export** utility or the **Tools→Export→Object** command.

The **export_recovery** utility is available for corrective actions on data that has ownership inconsistency that was not transferred using the SST protocol. You must also use this utility to correct failed offline site ownership transfer actions. The **export_recovery** utility also performs a real-time validation of the replica item's owning site before restoring site ownership to the local site. If the real-time validation fails due to some network error, the **export_recovery** utility prompts you to manually validate site ownership before continuing.

Transfer locks

Teamcenter 8.3 uses transfer locks during most transfers instead of the legacy modify lock. Transfer locks are not used during offline transfers. Unlike a modify lock, a transfer lock cannot be cleared by the **export_recovery** utility or the **clearlocks** utility even with the **-assert_all_dead** argument. Only the **ensure_site_consistency** utility can clear transfer locks. This prevents cases where objects that are being transferred are forcibly unlocked, which exposes them to the possibility of being modified while their ownership is being transferred.

If any transfer locks exist when you run the **clearlocks** utility with the **-assert_all_dead** or **-assert_dead** arguments, Teamcenter displays the following message:

```
Notice: There are transfer locks detected indicating active Multi-Site
```

```
transfer transactions. All transfers need to complete before the upgrade can
safely continue. Ensure that ensure_site_consistency is successfully executed
for any identified objects before running Clearlocks. Reference Multi-Site
System Administration section and Release Notes for additional information.
```

Teamcenter also displays this message if there are existing transfer locks during an upgrade to a new Teamcenter release.

If you get this message, wait for all transfer of ownership transactions to complete and run the **ensure_site_consistency** utility to complete all required recovery operations before rerunning the **clearlocks** utility with the **-assert_all_dead** or **-assert_dead** arguments again.

Fix mixed site ownership

Site ownership of an assembly and its related components and data can become mixed. This results in an **Item Already Owned** or **A replica of an object cannot be exported** error message when the assembly is exported or imported. You can fix this by:

- Converting the assembly item with mixed ownership to an item owned by the local site.
- Converting all objects in the assembly item to replicas.

Convert an item with mixed ownership to item owned by the local site

- Use the **export_recovery** utility **auto** mode. At a command prompt, type:

  ```
  export_recovery -mode=auto
  -item_id=corrupt-item-id -remote_site=site-name
  ```

  For information about the **export_recovery** utility, see *Utilities Reference*.

- If the **auto** mode fails to correct site ownership:

  1. Define the **TC_EXPORT_COPY** environment variable and set its value to **TRUE**.

     ```
     TC_EXPORT_COPY=TRUE
     ```

  2. Use the **item_export** utility to transfer site ownership to any site:

     ```
     item_export -item_id=item-id -owning_site=site-name
     ```

  3. Use the **export_recovery** utility **min** mode to read back the metafile output from the previous step:

     ```
     export_recovery -mode=min -dir=directory-name
     ```

  4. Delete the metafile output.

  5. Delete the **TC_EXPORT_COPY** environment variable.

- If site ownership is still not correct:

  – Use the **export_recovery** utility **auto** mode and specify the local site as the owning site:

  ```
  export_recovery -mode=auto -item_id=corrupt-item-id
     -remote_site=remote-site-owning-object_with_mixed_ownership
  ```

For example, if all objects in item are locally owned except a dataset that is owned by **Site1**, specify **-remote_site=Site1**. If there are other objects owned by other sites, this command must be repeated for each remote site.

Convert all objects in assembly item to replicas

1. Define the **TC_EXPORT_COPY** environment variable and set its value to **TRUE**.

   ```
   TC_EXPORT_COPY=TRUE
   ```

2. Use the **item_export** utility to transfer the assembly's site ownership to the true owning site:

   ```
   item_export -item_id=assembly-item-id -owning_site=true-owning-site-name
   ```

3. Delete the metafile output by the previous step.

4. Delete the **TC_EXPORT_COPY** environment variable.

If site ownership is still not correct, set the correct site in one of these ways:

- Use the **export_recovery** utility **auto** mode and specify the true owning site:

  ```
  export_recovery -mode=auto
  -item_id=assembly-item-id -real_site=true-owning-site-name
  ```

- In the rich client, select the assembly, choose **Tools→Export→Objects**, and export the assembly to the true owning site.

Working with log files

Teamcenter provides log files to assist you in troubleshooting your setup. At times you may want the most information possible in your log files to assist you during troubleshooting and at times large log files may be the cause storage or performance problems. Therefore, you must know how to change the way Teamcenter generates log files and how to interpret the log files.

Generating complete log files

To generate complete log files, you must define the following environment variables:

TC_Journalling=ON
TC_journaling=ON (Note the lowercase j and single l)
TC_Journal_Modules=ALL
TC_POM_JOURNALLING=N (Make sure it is N and not ON)
TC_TRACEBACK=ON

The log files extensions are:

- **.syslog**
- **.jnl**
- **.log**
- A **.mon** file is generated for My Teamcenter:

The file name is prefixed by the program name (IDSM, ODS, My Teamcenter and so on), followed by numbers which represent the PID number of the process.

Note

If the journal file, with the **.jnl** extension, is less that 2 MB, environment variables may not be defined or are incorrectly defined.

Interpreting the error stack

Generally, when an error is detected, an error stack is displayed consisting of a set of error codes and the corresponding meanings. The last error code displayed represents the ultimate cause of the problem and the preceding error codes represent how the error passes to higher levels of the code. The error stack is normally in the **syslog** file, and if you are using My Teamcenter, it displays in an error window.

When an error occurs while performing a remote operation, the error stack generated is very useful in debugging a problem. The following example shows an error stack from a failed remote import:

```
Error: 041010: Unable to import Item123
Error: 100107: Attempted function IDSM_start_remote_export at site
XYZ on host ABC
Error: 041131: Object "" has no export privilege.
```

There are several important points to remember from this example:

- Error code 41131 represents the root cause of the problem. In this case, an object at the remote site cannot be exported because it has no export privilege.

- In general, and not just for this example, error code 100107 indicates that the last error was detected at the remote site. That is, error code 41131 was detected while performing an operation at the remote site. Error code 100107 was detected at the importing site as well as all errors above it. Knowing where a specific error was detected is very important.

  Note

  If error code 100107 does not appear in the error stack, it is likely that the root cause of a problem was detected at the importing site during the local import phase.

- The quotations marks (**" "**) in the last error line normally contain the ID of the specific object that caused the problem. Because the error was detected at the remote site, the identity of the offending object is not available at the importing site.

  Although the first error line identified Item123 as the failed item, a specific subobject within the item is causing the problem. To locate the identity of the offending subobject, go to the owning site and do an interactive export by choosing **Tools→Export→Objects**. This reveals the ID of the offending object which can then be fixed to resolve the problem.

- The error stack that displays in the My Teamcenter: error window is also shown in the importing sites **syslog** file. Because the root cause of the problem was detected at the remote site, it is likely that the remote sites IDSM process generated its own **syslog** file that may contain more helpful information.

For more basic debugging techniques, see *Debugging remote import problems*.

Limiting the Oracle redo log size

During remote import, the Oracle redo logs can grow very quickly. You have some control over the grow rate of these logs. The import progress bar updates are logged in the background at intervals of *n* seconds, as determined by the **TC_RIMP_BG_prg_update_interval** site preference value, and when the import state changes.

To limit the growth of the redo logs, set the **TC_RIMP_BG_prg_update_interval** site preference to a numerical value greater than **5** (the default is **5**). This restricts the number of progress bar updates, so that input to the Oracle redo logs does not grow as quickly.

Postinstallation checklist

This section tells you what to look for after Multi-Site Collaboration is installed and before it is placed in operation. The information in this section provides a checklist that you can use to ensure a Multi-Site Collaboration system is properly installed. This does not mean that the checklist should be used only after Multi-Site Collaboration is installed. The checklist can be used to diagnose a Multi-Site Collaboration problem even when the system has been operational for some time.

Database entries

Choose **System Administration→Site Menu** and check that the following entries exist:

```
Site Name: Ods1
Site ID: 111111111
Site Node: node1
Object Directory Services button is enabled.

Site Name: Site2
Site ID: 222222222
Site Node: node2
Object Directory Services button is disabled.

Site Name: Site3
Site ID: 333333333
Site Node: node3
Object Directory Services button is disabled.
```

Note

The **Site Node** entry must be the server node where the ODS or IDSM daemon runs and not where the database server resides.

If the **Node** name of the database server is in the **Site Node** instead of the correct ODS or IDSM server node, your Multi-Site Collaboration user receives an error similar to the following:

```
RPC: Program not registered because the Oracle
server node would not have the ODS or IDSM setup
```

Site preferences

The site preferences for each site should contain the following Multi-Site Collaboration entries.

Site ODS1:

```
ODS_permitted_sites=
Site2
Site3
```

Site Site2:

```
IDSM_permitted_sites=
Site3
IDSM_permitted_transfer_sites=
```

Site3

Note

TC_transfer_area=
/users/tc_transfer_area

This preference is optional. If not defined, no site may transfer site ownership of objects owned by Site 2.

Note

TC_publishable_classes=
Item
Dataset
Form

If this site preference is not defined, the default value of **$TC_ROOT/transfer_area** is assumed. If the default directory does not exist, then CFI errors occur when performing Remote Import.

Note

The **TC_publishable_classes** preference is optional. If not defined, only items can be published.

Site Site3:

```
IDSM_permitted_sites=
Site2
IDSM_permitted_transfer_sites=
Site2
TC_transfer_area=
/users/tc_transfer_area
TC_publishable_classes=
Item
Dataset
Form
```

Note

The AM rule tree requires upgrading when you complete a version upgrade. If there are relatively few local changes then it is advised to merge these into the new rule tree. If there is a large local customization then it is advised to merge the new rule tree into the locally customized one.

Operating system directories and files

Note

The transfer area directory is defined by the **TC_transfer_area** preference.

The transfer area directory must be write-accessible to everyone. Any user performing remote import must be able to create or copy a file into the directory. Otherwise, you receive CFI errors when performing remote import. Metadata and operating system files are stored temporarily in this directory. Transfers attempted by users without write access to the directory causes fatal errors.

Note

This check is not needed for ODS-only sites such as ODS1.

Directory /etc: (UNIX only)

inetd.conf file

For IDSM server nodes, the **inetd.conf** file must contain an entry for the IDSM daemon:

Platform	Entry
HP-UX	**rpc stream tcp nowait infodba ${TC_ROOT]/bin/run_tc_idsm 536875586 1 run_tc_idsm**
SUN Solaris	**536875586/1 tli rpc/tcp nowait infodba ${TC_ROOT]/bin/run_tc_idsm run_tc_idsm**

Note

If this entry is missing, the IDSM daemon does not automatically start when a remote import request, or other IDSM requests, are sent to the site.

The **infodba** account is the default that is used by the installation procedures. It is possible that the account may have been intentionally changed by the customer (which is a valid change). The account used is referred to as the *IDSM user account* in the remaining sections.

File rpc

For IDSM server nodes, this file must contain an entry for the IDSM RPC listener as follows:

```
IDSM 536875586
```

When an IDSM server node is rebooted, this entry causes the system to create an IDSM RPC listener that you can see through the **rpcinfo -p** command. The listener is identified in the list as shown in the following example:

'536875586 1 tcp 49159'

Note

If this entry is missing, the IDSM daemon does not start automatically when a remote import request, or other IDSM requests, are sent to the site. Furthermore, the **rpcinfo -p** command does not show an IDSM RPC listener.

File run_tc_idsm

For IDSM server nodes, make sure that this file exists in this directory.

File run_tc_ods

For ODS server nodes, make sure that this file exists in this directory.

Schema compatibility

Before you attempt to import/export between two sites, make sure to check schema compatibility between the sites by running the **database_verify** utility. Using the output from the **database_verify** utility, make the schema of the sites compatible with respect to system objects, such as **Note Types**, **Dataset Types**, **Tools**, and **Release Status**. You prevent problems later by addressing this issue in the beginning.

Note

Schema incompatibility with respect to system objects would normally be manifested as a POM internal error.

Warning

Schema incompatibility with respect to class attributes can result in loss of data.

Common installation-related problems

Problems can occur as a result of incorrectly installing Multi-Site Collaboration. Problems can also occur after a system has been running Multi-Site Collaboration successfully, but certain changes were made (inadvertently or otherwise). For example, the **tc_profilevars** file may have been modified or the server node of an existing site may have been moved, but other sites were not notified. The troubleshooting information in the section can be applied to a brand new installation or to a site that is successfully running and suddenly develops a problem.

Unable to connect to an IDSM server error

Error code 100201 and its accompanying message indicates that the IDSM connection failed. There are several possible causes of this problem.

Follow the steps in the sequence they are presented to locate the cause of the problem. Proceed to the next step only if the current step does not reveal the cause of the problem. Assume that the error occurred while a user at Site 2 was trying to import an object from Site 3.

1. Check the site definition database entries at the requesting site.

The **Site Node** entry of the remote site should have the appropriate entry at the requesting site. In the example, if you check the requesting site (Site 2) for the site definition of the remote site (Site 3), the **Site Node** entry should show **node3**. If not, you must change the entry to **node3**. The user that received the error may have to restart his or her session before trying the remote import again.

Note

One of the most common installation errors when defining an IDSM site through the system administration site menu is putting the **Database Server Node** name as the **Site Node** entry. The correct **Site Node** entry is the server node where the IDSM daemon runs.

2. Check the network connection to the remote sites IDSM server node.

   Perform network-level tests to verify that the network connection between the sites is operational. For example, you can try to use ftp, rlogin, telnet, or any test you normally do without involving Multi-Site Collaboration. Check the network connection to **node3** from the node where the user runs Teamcenter. This is node is not necessarily node2 which acts as the IDSM server for Site 2. Also check the network connection between **node2** and **node3**.

3. Check the RPC connection to the remote sites IDSM server node.

   Using the **rpcinfo** utility at the requesting users node, perform RPC connection tests to the remote site, in this case node3. The command is as follows:

   rpcinfo -p node3

   The result should include an entry:

   ```
   536875586 1 tcp 32776
   ```

   Note

   The number 32776 is just an example and will be different in your case. However, the rest of the entry should be as shown.

   The Windows platform uses a graphical interface rather than a command line interface. Windows users must use the **Portmap Dump** menu option.

   This entry represents the RPC listener for the IDSM. You must check if the listener is ready:

   rpcinfo -T tcp node3 536875586 1

   or for some platforms:

   rpcinfo -t tcp node3 536875586 1

   Note

   The ability to check if the listener is ready is not available on Windows.

4. **Check the IDSM start-up files at the remote sites IDSM server node.**

   You must log in to the remote site and check the files involved in the startup of the IDSM server. These files include those in the *TC_ROOT*/**bin** directory

named **inetd.conf**, **rpc**, and **run_tc_idsm**. Review the preceding text for what these files should contain.

Detecting problems with the **inetd.conf** and **rpc** files is straightforward. The **run_tc_idsm** file can require more investigation. If the problem has not been resolved at this point, you must check the **run_tc_idsm** file and the scripts it calls.

5. **Check the run_tc_idsm script.**

   - If recent changes have been made to this file, check to make sure the changes are correct.

   - Test if the script is getting invoked. A simple test is to edit the script and add similar to:

     ```
     echo $TC_DATA > /tmp/test.tmp
     ```

     Try the Multi-Site Collaboration operation again. Check if the **/tmp/test.tmp** file was created and has a valid value for **TC_DATA**.

     If the file was not created, the **run_tc_idsm** script is not getting executed and you must go back and double check everything you have done so far.

     If the file **/tmp/test.tmp** was created, then make sure that the value of **TC_DATA** is the correct value. If not, edit the **run_tc_idsm** file to set the correct value for **TC_DATA**.

     – Log on to the operating system as the administrator user, or the IDSM user account specified in the **inetd.conf** file.

       Did you notice any problem?

       Are the environment variables set correctly?

       Try running Teamcenter using the same IDSM user account. Any problems?

       Does Teamcenter log on to the right database?

     – Check the **tc_profilesvar** file in the **TC_DATA** directory for any recent changes that may affect the IDSM.

       Note

       A common error made in the **tc_profilesvar** script is adding some entries that are needed only for interactive users but not for background processes like the IDSM. If you have such entries, make sure you make their execution conditional with an if interactive condition.

6. **Check IDSM syslog**.

   If the IDSM server was actually started but died without giving any error messages, it would very likely create a **syslog** file. The **syslog** file is created in the directory defined by the **TC_TMP_DIR** environment variable, which is normally assigned to **/tmp** or **/var/tmp**. It would have a name of **IDSM***nnnn***.syslog** where *nnnn* is the process pid.

If the **syslog** file exists, then check the contents and look for errors. Start from the bottom of the file when looking for errors because the file might contain some error messages that the system ultimately recovers from.

If the error is something you can fix, make the necessary corrections. If not, report the problem but before doing so, perform the next step to generate more debugging information.

7. **Generate Log files for debugging.**

   Edit the **run_tc_idsm** file and right before the very last line where the IDSM daemon is started up using the **exec** command, add the lines to define the environment variables described previously for generating complete log files.

   After making the changes, try the Multi-Site Collaboration operation again. Make sure to try the Multi-Site Collaboration operation several times. For example, if you are importing an object, try the import a couple of times. The reason for this is the journaling mechanism buffers the last few blocks of journal information; by doing the operation multiple times, the part with the errors is written to the log.

   Gather all of the log files that were generated and send them in with your problem report.

Unable to connect to an ODS server error

Error code 100101 and its accompanying message indicates that the ODS connection failed. There are several possible causes of this problem. Perform the steps in the indicated order to pinpoint the cause of the problem; proceed to the next step only if the current step does not reveal the cause of the problem. This example assumes the error occurs when a user at Site 2 attempts to publish an object to ODS1.

1. **Check the site definition database entries at the requesting site.**

   The **Site Node** of the remote site should have the appropriate entry at the requesting site. In our example, if you check at the requesting site (Site 2) for the site definition of ODS1, the **Site Node** entry should show **node1**. If not, you must change the entry to **node1**. Note that the user who received the error may have to restart his or her session before trying the remote import again.

   Note

   One of the most common installation errors when defining an ODS site using the system administration site menu is putting the node name of the database server as the **Site Node** entry. The correct **Site Node** entry is the server node where the ODS daemon runs.

2. **Check the network connection to the ODS server node.**

   Perform some network-level test to verify that the network connection between the sites is operational. For example, you can try to use **ftp**, **rlogin**, **telnet**, or any test you normally do without involving Multi-Site Collaboration. Continuing our example, check the network connection to node1 from the node where the user runs Teamcenter. Note that this is not necessarily **node2**, which acts as the IDSM server for Site 2. Also check the network connection between **node2** and **node1**.

3. **Check the RPC connection to the ODS server node.**

Using **rpcinfo** at the requesting users node, perform RPC connection tests to the ODS sites server node, in this case node1. The command is as follows:

```
rpcinfo -p node1
```

The result should include the 2 entries:

```
536875585 1 udp 32774
536875585 1 tcp 32775
```

Note

The numbers 32774 and 32775 are examples and will be different in your case. However, the rest of the entries should be as shown.

The Windows platform uses a graphical interface rather than a command line interface. Windows users must use the **Portmap Dump** menu option.

The **tcp** entry represents the RPC listener for the ODS. You must check if the listener is ready:

```
rpcinfo -T udp node1 536875585 1
```

or

```
rpcinfo -T tcp node1 536875585 1
```

for some platforms.

The response from **rpcinfo** should be:

```
program 536875585 version 1 ready and waiting
```

If you do not see the entries in the list returned by **rpcinfo -p** or the response from **rpcinfo -T** is not as described, there is a problem at the remote site. The problem can be narrowed down with the next steps which require you to log on to the ODS server node, in our example **node1**.

Note

The ability to check if the listener is ready is not available on Windows.

4. **Check that the ODS server process is running.**

You must login to the ODS server node, node1, and perform some checks.

Unlike the IDSM daemon which is started up on demand, the ODS daemon is started up when the ODS server node is rebooted. Check that the ODS daemon is running as follows:

```
ps -ef | grep ods
```

This command lists all processes in the system and then display the ones with the **ods** string. This should show at least 2 lines: one with the string **$TC_ROOT/bin/run_tc_ods** and the other with the pathname of the ODS executable, the expansion of **$TC_BIN/ods**.

If either line is missing, the ODS daemon is not running and must be restarted. Preferably, the system should be rebooted but if this causes problems with other users, then use the **$TC_ROOT/bin/run_tc_ods** script.

After restarting the ODS, check to see if the daemon is running. If so, try the Multi-Site Collaboration operation again.

If the ODS daemon is not running, it is terminating prematurely and you must continue with the next step.

5. **Check the run_tc_ods script for possible problems.**

   The **run_tc_ods** script starts up the ODS daemon. Before doing so, it sets up environment variables within itself by calling **$TC_DATA/tc_profilevars**, which the ODS uses. For this reason, it is important to check if the correct environment variables are being set.

   If recent changes have been made to this file, then check to make sure the changes are correct.

   Log on to operating system as the administrator user. Did you notice any problem? Are the environment variables set correctly? Try running Teamcenter as the administrator user. Any problems? Does Teamcenter log on to the right database?

6. **Check the ODS syslog file.**

   If the ODS daemon continues to terminate prematurely after trying the previous fixes and restarting it, check for a **syslog** file created when it terminates. The **syslog** file is created in the directory defined by the **TC_TMP_DIR** environment variable which is normally assigned to **/tmp** or **/var/tmp**. It has a name of **ods***nnnn***.syslog** where *nnnn* is the process pid.

   If the **syslog** file exists, check the contents and look for errors. Start from the bottom of the file because the file may contain error messages that the system ultimately recovers from.

   If the error is something you can fix, make the necessary corrections. If not, report the problem but before doing so, perform the next step to generate more debugging information.

7. **Generate ODS log files for debugging.**

   Edit the **run_tc_ods** file and right before the last few lines where the ODS daemon is started up, add the lines to define the environment variables described above for generating complete log files.

   After making the changes, try the Multi-Site Collaboration operation again. Make sure to try the Multi-Site Collaboration operation several times. For example, if you are publishing an object, try publishing a couple of times. The reason for this is the journaling mechanism buffers the last few blocks of journal information; by doing the operation multiple times, the part with the errors is written to the log file.

   Gather all of the log files that were generated and send them in with your problem report.

ODS returns an ACS or licensing error

The ODS daemon attempts to obtain an ODS server license on the first Multi-Site Collaboration request that it gets. So it is possible for the ODS daemon to actually stay up and running after it is started without getting a license. Once it gets an ODS server license, it holds on to that license until it is terminated.

In order to avoid using too much memory, the ODS daemon restarts itself automatically every 1000 requests; this is the default value which can be changed

in the **run_tc_ods** script. In the process of restarting itself, the ODS daemon unallocates the ODS server license. Upon restarting, it is in a state where it has not received a Multi-Site Collaboration request and therefore it has not allocated an ODS server license; it allocates the license upon getting the next incoming request.

- It is possible to encounter this problem immediately after the system is rebooted or after the ODS has been operational for sometime. There are several possible causes of this problem:
  - The license server, ACS or Flex, may not have an ODS server license.
  - Another ODS server may have allocated the license ahead of your ODS.
  - The ODS daemon may have been inadvertently terminated and did not get an opportunity to release the ODS server license. Check if the ODS server license is allocated by the license manager.

If the problem persists, perform step 6, Check the ODS syslog to debug the **syslog** file and step 7, Generate ODS log files to generate complete log files in *Unable to connect to an ODS server error*.

Not logged on to expected site error

The end user sees this problem as error code 100106, for ODS operations, or 100213, for IDSM operations, with an accompanying message that the ODS or IDSM server is not logged on to the expected site. There are several possible causes of this problem. Perform the following steps in the order indicated to find the cause of the problem; proceed to the next step only if the current step does not reveal the cause of the problem.

The **Site Node** database entry contains the ODS or IDSM server node name and not the database server node. The ODS and IDSM daemons use the **TC_DB_CONNECT** environment variable to determine which database to use. This environment variable is defined when the **$TC_DATA/ tc_profilevars** is sourced in the **run_tc_ods** and **run_tc_idsm** scripts. Incoming requests to the daemon contain the identity of the server site, that is, the site to apply the request to. If the identity of the server site as supplied by the requesting site does not match the value of **TC_DB_CONNECT**, a mismatch occurs.

The requesting site determines the identity of the server site as follows:

- For ODS requests, **Publish/Unpublish** and **Find Remote**, the server site is usually obtained from the **ODS_site** preference, unless the user is publishing to or unpublishing from a specific ODS.
- For **Import/Export** requests, the server site is determined from the owning site attribute as contained the publication record.

  Note

  If the publication record does not reflect the current owning site, Multi-Site Collaboration can determine the current owning site by starting with the current information in the ODS and determining the current owner that is used to identify the server site.

To determine the cause of the problem:

1. Check the **ODS_site** preference at the requesting site. Make sure the entry is the default ODS.

2. Check the **Site Node** database entries at the requesting site. The correct entry is the name of the node running the ODS or IDSM daemon and not the database server node.

3. Check **run_tc_ods** or the **run_tc_idsm** script as appropriate to make sure the **TC_DATA** environment variable has the correct value. Temporarily adding a statement, such as **echo $TC_DATA > /tmp/test123.tmp** to the script, helps determine if the correct value is being used.

4. Check **tc_profilevars** script in **TC_DATA** to make sure the **TC_DB_CONNECT** environment variable is being assigned the correct value. Temporarily adding a statement, such as **echo $TC_DB_CONNECT > /tmp/test123.tmp** at the right point in the script, helps determine if the correct value is being set.

5. If inspecting the above scripts does not reveal the cause of the error, log on to the operating system as the administrator user, or the account specified in **inetd.conf** in the case of IDSM, and check the values of **TC_DB_CONNECT** and **TC_DATA**. The value of **TC_DB_CONNECT** must match the current owning site of the object being imported, if importing, or the default ODS, if publishing/unpublishing.

If these steps do not resolve the problem, generate complete log files as described in *Unable to connect to an ODS server error*. Send the log files in with your problem report.

Common import/export problems

In most cases, when a remote import operation fails, it displays error codes and error messages which are sufficient to help the user resolve the problem. For example, the error message may indicate the lack of privilege to import a given remote object; or when reimporting an object, the error can indicate that the object is in use. In these cases, the resolution of the problem is obvious. These types or error conditions are not the subject of this section. Rather, this section discusses errors where the resolution is not straightforward, even for a system administrator who is trying to help a user resolve an import problem.

It is important to describe a general procedure that can be used in debugging many import/export problems. At several points in the document, you are asked to perform this general procedure as the first step in resolving an import/export problem.

Note

This general procedure can also be used for debugging import/export problems encountered while using interactive My Teamcenter import/export commands and the **item_export** and **item_import** command line utilities.

Debugging remote import problems

The general procedure is based on the Multi-Site Collaboration remote import operation performing the same basic operations involved in performing manual

import/export operations. When the remote import or **Commands→Import Remote** operation returns an error where the resolution is not obvious, follow these procedures:

1. Define the environment variables described previously for generating complete log files before running the rich client to perform interactive import/export.

2. At the owning site, choose **Commands→Export→Objects** on the same object that failed and with the same import/export options specified in the remote import.

   If the interactive export fails, the problem is on the export portion of the operation. Note the error messages generated; in most cases, the error messages are more informative than what is returned by remote import and help lead to a quick resolution of the problem.

   If you cannot resolve the problem based on the error messages, gather all the log files together and send them to Siemens PLM Software with your problem report.

3. If the export succeeds, send the export directory and all its contents to the importing site using FTP or other similar means.

4. At the importing site, after defining the environment variables for generating complete log files, import the data using **Utilities→Files→Import→Objects**. If the import fails, the problem is in the import portion of the whole operation. Note the error messages generated; in most cases, the error messages are more informative than what is returned by remote import and helps lead to a quick resolution of the problem.

   If you cannot resolve the problem based on the error messages, gather all the log files together and send them in with your problem report.

5. If the import succeeds, the remote import problem is likely in the networking portion of Multi-Site Collaboration which involves the IDSM and ODS server and the network and RPC environments they operate in.

   Network and RPC problems are normally reported as connection or RPC-related error messages that are dealt with separately in this guide. Assume the remote import problem is due to the operation of the IDSM server.

6. Check the IDSM **syslog** file. The **syslog** file is created in the directory defined by the **TC_TMP_DIR** environment variable, which is normally assigned to **/tmp** or **/var/tmp**. It has a name of **IDSM***nnnn***.syslog** where *nnnn* is the pid of the process.

   If the **syslog** file exists, then check the contents and look for errors. Start from the bottom of the file when looking for errors because the file may contain error messages that the system ultimately recovers from.

   If the error is something you can fix, make the necessary corrections. If not, report the problem, but before doing so, perform the next step to generate more debugging information. Also do this if you cannot find the IDSM **syslog** file.

7. Generate IDSM log files for debugging. Edit the **run_tc_idsm** file, and before the last line where the IDSM daemon is started using the **exec** command, add the lines to define the environment variables described previously for generating complete log files.

After making the changes, try the remote import again. Make sure to try the remote import several times as the journaling mechanism buffers the last few blocks of journal information; by doing the operation multiple times, the part with the errors are written to the log.

Gather all of the log files that were generated and send them in with your problem report.

Invalid directory contents error

When the import returns an invalid directory contents error, this is normally caused by an outdated POM transmit file.

What exactly is a POM transmit file? Assume that we are exporting from Site 2 into Site 3. The result of exporting from Site 2 is an export directory with a metafile that contains the exported objects. The whole export directory is brought over to Site 3 for import. For Site 3 to understand the contents of the metafile, it needs to know something about the schema of Site 2.

What are the classes of the objects in the metafile? What are the attributes of the classes? This is where the POM transmit file comes in. It contains a description of the classes, their attributes and other important schema information at the exporting site. Using this schema description, the importing site is then able to interpret the data in the metafile.

Note

Do not confuse the POM transmit file with the POM schema file which is pointed to by the **POM_SCHEMA** environment variable. While both contain information about a site schema, the transmit file is geared towards the import/export, and also archiving, of objects. For this reason, a site can have several transmit files in the **POM_TRANSMIT_DIR** directory, one for each stage of the site schema evolution, so that objects exported or archived at a specific stage can be imported even after the schema has evolved.

For the **item_import** utility and the rich client import/export operations to work, a current transmit file of the exporting site must be available at the importing site **POM_TRANSMIT_DIR** directory. The transmit file is placed in the directory manually as a routine part of site maintenance.

When you perform a remote import operation, Multi-Site Collaboration checks for the exporting sites transmit file at the importing site **POM_TRANSMIT_DIR** directory.

When remote import returns an *invalid directory contents* error:

1. Log on to the exporting site and check if its transmit file exists and is up-to-date.

   The transmit file of a site is a file with the extension **.om_sch** and a name that contains the numeric site id. For example, Site 2's transmit file has the string **_22222222_** in its name. There can be several of these in the **POM_TRANSMIT_DIR** directory. If so, the one with the latest creation date is used by the system.

   The transmit file of a site is a file with the extension **.om_sch** and a name that contains the numeric site id. For example, Site 2's transmit file has the string **_22222222_** in its name. There also could be several of these in the **POM_TRANSMIT_DIR** directory. If so, the one with the latest creation date is used by the system.

If the transmit file exists, make sure it is up-to-date. You verify this by comparing its creation date and time with that of the POM schema file. The transmit file must have a later date and time than the POM schema file.

Note

The use of the OS-level creation date and time is not guaranteed accurate. The real timestamp is the cryptic string that is part of the transmit file name. To ensure you have most up-to-date transmit file, temporarily rename the file by adding a **.save** extension and then regenerate the transmit file as described in the following steps. After doing so, rename the **.save** file back to its original name.

2. If the POM transmit file is out-of-date, generate a new one by entering the following command:

```
$TC_BIN/install -gen_xmit_file infodba
   infodba password dba
```

Note

You must distribute the new transmit file to the other sites, especially if you plan to use **item_export/item_import** utilities or the rich client import/export commands. Although it is not necessary to do so when you use Multi-Site Collaboration exclusively to perform import, distributing the transmit file to the other sites is still important for debugging Multi-Site Collaboration problems. For instructions about debugging remote import problems, see *Debugging remote import problems*.

3. Check the **tc_profilevars** file to make sure the **POM_TRANSMIT_DIR** environment variable points to the correct directory.

Caution

When IDSM or ODS is configured to run as a Windows service, you must use a UNC formatted path for the **POM_TRANSMIT_DIR** variable. If you use a network drive (mapped) letter in this variable, the service is not able to locate the directory to read the required files.

If the invalid directory contents error is accompanied by a POM internal error, see *POM internal error*.

POM internal error

When importing, a POM internal error message usually indicates there is a schema discrepancy between the exporting and importing sites. The schema discrepancy that causes this error is normally associated with types, such as **Dataset** which are defined at the exporting site, but not the importing site.

A manual check of items like **Note** types, **Form** types, **Dataset** types, and **Tools** is usually sufficient to solve the given problem. However, Siemens PLM Software recommends that you run the **database_verify** utility to check all the different system objects and types in order to avoid future problems.

Item has inconsistent site ownership

When an error or a system crash occurs in the middle of an import/export operation that involves transfer of site ownership, the item can have an inconsistent site ownership. Inconsistent ownership exists when the item is owned by one site and some revisions or attachments are owned by another site. Any attempt to import/export such an item results in error 41121: `A replica of an object cannot be exported`.

This error can occur on the master copy or on a replica of an item. In either case, correct the problem as follows.

- For online transfers, use the **ensure_site_consistency** utility to correct the inconsistency and clear transfer locks.

  For more information, see *Recovering data due to failed transfer of ownership*.

- For offline transfers or transfers that involve legacy data (such as during a upgrade) use the **export_recovery** utility to correct the problem and clear modify locks. If there are existing transfer locks, run the **ensure_site_consistency** utility before using the **export_recovery** utility.

For additional information on using these utilities, see the *Utilities Reference*.

After running one or both of these utilities, the site ownership of all objects in the item should be consistent. If it is a master copy, try exporting it without site ownership transfer. If the export succeeds, this confirms the success of the recovery.

Configure a Teamcenter UTF-8 execution environment on UNIX

If you are running the IDSM server process on a UNIX platform, the **data_share** utility may fail with an error that states it cannot find an item when the **item_id** attribute contains non-English characters. This occurs on UNIX systems when the IDSM daemon is not started with identical Teamcenter execution environment settings. By default, the IDSM server process is started by the UNIX **idsminetd** daemon in the C locale.

To establish a UNIX Teamcenter UTF-8 character set execution environment, the following variable settings must be added to the **run_tc_idsm.sh** IDSM startup script file located in the *TC_BIN* directory:

```
LANG=en_US.UTF-8
LC_ALL=en_US.UTF-8
```

Item ID duplication

Duplicate item IDs cannot be used within a Multi-Site Collaboration federation. Item IDs must be unique. Multi-Site Collaboration includes centralized item ID functionality you can use to:

- Identify existing cases of item ID duplication.
- Resolve existing cases of item ID duplication.
- Prevent further duplication of item IDs.

Identifying item ID duplication

You can identify existing cases of duplicate item IDs using the **data_share** utility. Use the utility to find duplicate item IDs, and to register or unregister defined item IDs from the Central Item Registry.

The item ID search accepts wildcards and can be constrained to search by creation date of the item. The utility returns such information about suspected duplicate item IDs as the unique identifier, owning site, item description, and so forth.

For additional information on the use of this utility, see the *Utilities Reference*.

> **Note**
>
> Cases of duplicate item IDs result from one of six possible causes. For information on resolving cases of duplicated item IDs once you have identified them using the **data_share** utility, see *Resolving item ID duplication*.

Resolve existing cases of item ID duplication by performing the recommended solution corresponding to the cause that occurred at your site.

Prevent future cases of duplicate item IDs by implementing the Central Item Registry at Multi-Site Collaboration sites.

> **Note**
>
> Central Item Registry checks for duplicate IDs when **ASSIGN** is used, if that preference is set. For more information, see the *Preferences and Environment Variables Reference*.

Resolving item ID duplication

Identify existing cases of duplicate item IDs using the **data_share** utility. Cases of duplicate item IDs result from one of the following possible causes.

Resolve existing cases of item ID duplication by performing the recommended solution corresponding to the cause that occurred at your site.

Prevent future cases of duplicate item IDs by implementing the central item registry at Multi-Site Collaboration sites.

Resolution scenarios

The following scenarios define the possible causes of item ID duplication and recommend solutions:

Scenario 1

Items are identical except for having different unique IDs. Each is considered to be owned by their respective sites.

- **Likely Cause:**

  The **ug_import** utility was used to import the same item into two different sites.

- **Recommended Solution:**

  Declare one item the master copy. Import the other item with transfer ownership for the forms and datasets that do not exist on the master item. Create new revisions on the master item for the objects imported.

Scenario 2

Items are different in every way, including having different unique IDs.

- **Likely Cause:**

  Items were created separately. Nothing prevented the two items from sharing the same item ID.

- **Recommended Solution:**

  Change the item ID of one of the items.

Preventing item ID duplication

Prevent duplicating item IDs when creating new items by using the central item registry. When this functionality is enabled, item IDs are checked against the registry to ensure the item ID does not exist with the Multi-Site Collaboration federation. The registry is a **ItemIdRegistry** table containing the set of all item IDs created within the Multi-Site Collaboration federation.

Enabling the central item registry

Enable the central item registry functionality by setting the following site preferences in the preference XML file:

- Set the **ITEM_id_registry** site preference to enable the Central Item Registry functionality. All other Central Item Registry preferences are ignored unless this preference is enabled.
- Set the **ITEM_id_registry_site** site preference to define the site of the registry.
- Set the **ITEM_id_always_register_on_creation** site preference to automatically register item IDs when creating an item. If this preference is disabled, new items must be manually registered.
- Set the **ITEM_id_allow_if_registry_down** site preference to determine whether item creation fails if item ID registration is required but the central registry is unavailable.
- Set the **ITEM_id_unregister_on_delete** site preference to determine if item IDs are automatically unregistered when items are deleted or the item ID is changed.

Refer to the *Preferences and Environment Variables Reference* for additional information regarding these preferences.

Registering item IDs in the central item registry

Register item IDs within the registry one of two ways:

- Register the item IDs of existing items by selecting the item and using the **Tools→Multi-Site Collaboration→Item ID Registry→Register Item ID** menu option.

- Automatically register the IDs of new items during item creation by enabling the **ITEM_id_always_register_on_creation** site preference.

Windows platform notes

Most of Multi-Site Collaboration is platform independent. The portions that are platform dependent involve the RPC-related aspects the ODS and IDSM and are the subject of this section.

On Windows, the ODS is run as a Windows service. Services in Windows are analogous to daemons on UNIX and are normally started at boot up time.

The IDSM on UNIX depends on a daemon to listen for requests and launch the IDSM server. On Windows, an IDSM front-end service has equivalent functionality. This front-end service launches the real IDSM server on demand.

How to determine if the services are running

1. Launch the control panel and select the **Services** applet.

2. Scroll down to the **NobleNet Portmapper** service; the status should be **Started**.

3. Scroll up to the **IDSM** service. If the services do not appear, Multi-Site Collaboration must be installed and configured before proceeding.

4. The status of the two services should be **Started**. If not, highlight each service and click **Start**.

5. If the services still do not start, check if the correct user is specified as the owner of the service. Highlight a service and click **Startup**.

6. Check if the correct user account is being used; see the following **This Account** entry. If the correct account is shown, reselect the user from the user browser, the ellipses, and enter the password again.

7. At this point, the services should have started. If not, check the **%TC_BIN%** directory to determine if the **run_tc_idsm.bat** and **run_tc_ods.bat** files exists and try manually running the batch files. This can generate a useful error or a **syslog** file.

8. Perform the RPC connection tests described in *Common installation-related problems*, for both the ODS and the IDSM. These tests use **rpcinfo**.

   - Follow the steps in *Unable to connect to an ODS server error*, to perform the RPC connection test for an ODS server.

   - Follow the steps in *Unable to connect to an IDSM server error*, to perform the RPC connections test for an IDSM server.

Note

On Windows, Teamcenter is shipped with a graphical version of the **rpcinfo** program located in the **%TC_BIN%** directory.

Part III *Using Multi-Site Collaboration*

Note

Your system administrator must have previously set up and configured Multi-Site Collaboration before you can use these features and functions.

Chapter

10 Best practices using Multi-Site Collaboration

Chapter

10 Best practices using Multi-Site Collaboration

To avoid performance or operational problems:

- **Publish high-level objects.**

  Publish high-level objects such as items, not individual low-level objects such as forms and datasets. When you publish an item, all underlying objects are imported when the item is imported.

- **Specify at least one target site when exporting an object.**

  When exporting an object, you must specify at least one target site. Otherwise, the export operation produces an export file that cannot be imported.

Chapter

11 Publishing and unpublishing

Chapter

11 Publishing and unpublishing

Publishing an object makes that object available to other sites; unpublishing an object reverses the procedure; the object is only accessible by the local owning site.

Your system administrator defines a default ODS for your entire site. You cannot change the default ODS and you are expected, in most cases, to publish to the default ODS. The system administrator may also have defined a list of ODS publication sites that you can use to publish to multiple ODS sites, at the same time. Consult with your system administrator for additional information.

Note

When sharing form data between multiple site, ensure the form storage class is properly defined at the importing site and is compatible with the form storage class at the exporting site.

Participating sites in a distributed network must have some reliable way of controlling which data they want to share with the rest of the network. With Multi-Site Collaboration, you can publish and unpublish objects.

- *Publishing* an object makes that object available to other sites. When you publish an object, a publication record is created in the ODS that can be read and searched by other Teamcenter sites. Until you publish an object, it can only be seen by the local owning site, other sites are not aware that it exists.

  To view the objects in a folder that are currently published, select the folder and check the **Status** column in the **Details** table.

  To see if an item is published, right-click the item and choose **Properties**. Click **All** in the **Properties** dialog box. If an item is published, Teamcenter displays the ODS sites where it is published in the **Published To** box.

- *Unpublishing* an object reverses the procedure, the object is only accessible by the local owning site.

  Note

  Your administrator has defined a default ODS for your entire site. You cannot change this and you are expected, in most cases, to publish to the default ODS. The administrator may also have defined a list of ODS publication sites that can be used to publish to multiple ODS sites, at the same time. Consult with your administrator for additional information.

Publish an object

1. Select the object to publish.

2. Choose **Tools→Multi-Site Collaboration→Publish** and:

To	Do this
Publish to the default ODS site with the default selection rules.	Choose **To Default ODS**.
Publish to select ODS sites.	1. Choose **To Default ODS…** 2. Click the **Select Site** button  . 3. In the **Site Selection** dialog box, add or remove sites to publish to in the **Selected Sites** list. 4. Click **OK**.
Publish to the default ODS with specific selection rules.	1. Choose **To Default ODS…** 2. In the **Publish To Default ODS** dialog box, click the **Explore Selected Component(s)** button  . 3. In the **Explore** dialog box, select the desired **Selection Rules**. 4. Click **OK**.
Publish to all ODS sites in the publication list with the default selection rules.	1. Choose **To ODS Publication List**. 2. In the **Publish to Publication List** dialog box, click **Yes**.
Publish to all ODS sites in the publication list with specific selection rules.	1. Choose **To ODS Publication List**. 2. In the **Publish to Publication List** dialog box, click the **Explore Selected Component(s)** button  . 3. In the **Explore** dialog box, select the desired **Selection Rules**. 4. Click **OK**.

Unpublish an object

1. Select the published object.
2. Choose **Tools→Multi-Site Collaboration→Unpublish** and:

To	Do this
Unpublish from the default ODS site.	Choose **From Default ODS**.
Unpublish from the default ODS sites or selected ODS sites.	1. Choose **From Default ODS...** 2. Click the **Select Site** button  . 3. In the **Site Selection** dialog box, add or remove sites to unpublish from in the **Selected Sites** list. 4. Click **OK**. 5. Click **Yes**.
Unpublish from all ODS sites without further user actions.	Choose **From All ODS Sites**.
Unpublish from one or more specific ODS sites.	1. Choose **From Specific ODS Site(s)**. 2. In the **Site Selection** dialog box, add or remove sites to unpublish from in the **Selected Sites** list. 3. Click **OK**. 4. Click **Yes**.

Multi-Site Collaboration publish privilege

The **PUBLISH** privilege controls both the publishing and unpublishing of objects. You must have **PUBLISH** privilege on an object to publish or unpublish an object. Your administrator defines the rules that determine who has publishing privileges on objects.

Typically, the owner of the object automatically gets publishing privilege. If you do not have the privilege to publish an object, an attempt to publish or unpublish the object returns an error. Check with your administrator about the Access Manager rules that control publishing privileges.

Chapter

12 Object protection and ownership

Chapter

12 Object protection and ownership

In a normal noncollaborative environment, the ownership and protection of objects is straightforward and generally transparent to users. However, in a collaborative environment, a level of complexity is greatly increased in order to extend object protection across an entire network.

Site ownership

In addition to the familiar concepts of owning user and owning group, Multi-Site Collaboration uses the concept of an *owning site*. The owning site is the site where the master object of an object resides. It is the only site where the object can be modified. It is the only site where you can obtain a replicated copy of the master object. The owning site is a property of any object and the owning site can be found in the **Properties** dialog box. When an object is replicated by a remote site, the owning site property will go along with it. However, other aspects of access control may vary for each replica according to the environment of the replicating (that is, remote) site.

Access control on replica data

All replicas are read-only objects, regardless of whether the site uses rules-based or object-based protection. When an object is replicated, the owning user and owning group for the replica are determined as follows:

- If the owning user and owning group of a master object are both defined at the importing site, the imported copy (replica) will be owned by this user and group following the import, that is, the ownership is fully preserved.
- If either the owning user or owning group of a master object is not defined at the importing site, the imported copy (replica) will be owned by the user performing the import; the owning group will be that user's current group at the time of the import.
- If the value of the **TC_retain_group_on_import** site preference is **TRUE** and the owning group is defined at the importing site, original owning group will be preserved.

  Note

  These rules also apply when site ownership is transferred from one site to another.

When an object is exported from a site using traditional object-based protection (that is, not using rules-based protection) and imported into a site using rules-based object

protection, access controls at the importing site apply (subject to the limitation that remote objects are always read-only). This is true regardless of whether site ownership is transferred or not.

Site autonomy

Multi-Site Collaboration intentionally imposes as few restrictions and limitations on autonomous site activity as possible. This includes object protection and ownership. Sites are not required to define users from other sites in their database and each site is free to choose the object protection scheme (object-based or rules-based) used at their site. Furthermore, if rules-based object protection is used, each site is free to define.

Site unity

Siemens PLM Software recommends that, if possible, all sites use rules-based object protection and define similar rules so that access to shared objects is uniform across the entire Multi-Site Collaboration network. Furthermore, defining a consistent set of users for all sites, though impractical for some enterprises, is recommended whenever possible.

Chapter

13 Remote import and export options

Chapter

13 Remote import and export options

Use the **Import Remote Options** and the **Export Options** dialog boxes to set the options for importing remote objects and exporting objects to other sites.

These options enable you to control:

- Transfer of site ownership
- Whether to perform a remote import operation or an export objects operation in the background or foreground
- Which relationships to include or exclude
- Which item revision to import/export
- Whether or not to include assembly components
- Import/export report options
- Synchronization and notification options

Each option in the dialog box has a default value. The system retrieves the default values from the preference file or these values are predetermined if there are no default values set in the **Preference** file.

Transfer options	General tab
Transfer Ownership	Set this option to transfer site ownership to the target site. When the this option is not set, your site retains ownership. If you transfer an item revision with a sequence, its sequence manager is also transferred. The **TC_ownership_export** preference controls the default value of this option. **Note** Siemens PLM Software recommends that you leave the default setting for this option to **unset**.

Transfer options	General tab
Perform Import/Export in Background	**Remote Import** If selected during a remote import, set this option to execute the **Remote Import** operation in the background so you can continue to use your workspace session while the import/export operation takes place behind the scene. You are allowed to import in background only one selected object at a time so it is recommended that you use this option for importing an assembly. While the background operation takes place, you can perform other Multi-Site Collaboration operations. Even with the multiple simultaneous Multi-Site Collaboration operations, only one Multi-Site Collaboration user license is used. **Remote Import Progress** indicators are visible in the remote import background mode. When the background process completes, a dialog box appears to inform you of the completion status. If the import is successful, the imported object is placed in the **Newstuff** folder. **Note** Select this option if want to continue using Teamcenter while the **Remote Import** operation runs. **Object Export** If selected during an interactive object export, this option executes the export operation in the background so you can continue to use your Teamcenter session while the import/export operation is performed. The export output is placed in the directory specified by the **TC_background_object_export_dir** site preference. If this preference is not defined, the default setting is the **/tmp** directory. When the export is complete, an e-mail is sent to the user at the e-mail address defined in the database. The e-mail notifies the user of the success or failure of the operation, and the location of the export data. If the user has selected the **Generate Import/Export Report** option, the report is included in the e-mail. While the background operation takes place, you can perform other Multi-Site Collaboration operations. Even with multiple simultaneous operations proceeding, only one Multi-Site Collaboration user license is used. **Note**

Transfer options	General tab
	During this operation, the export directory and its contents are given operating system-level access protection based on the protection mask of the **tcserver** process. Typically, the **tcserver** process uses the **infodba** setting, thus the export files are normally accessible to users with **infodba** and/or privileged access. This protects the export directory from unprivileged users because the export files are intended to be placed on backup media for shipment to other sites.

Item options	Description
Include All Revisions	Set this option to export all revisions. When transferring site ownership, this is the only option available.
Latest Revision Only	Set this option to export the latest revision regardless of whether it is a working or released revision.
Latest Working Revision Only	Set this option to export only the latest working (such as nonreleased) revision.
Latest Working/Any Release Status	Set this option to export the latest working revision, if any; if no working revision, the latest released revision with any release status is exported.
Latest Any Release Status	Set this option to export the latest released revision with any release status.
Selected Revision(s) Only	Set this option to export only the revision(s) selected in the workspace. This option is not valid for **Remote Import**.
Specific Release Status Only	Select this option to export only the latest revision with the given release status selected from the list. This is available only in the rich client.

The following options allow you to filter out workspace objects that you want to include or exclude in the import/export operation. These are normally applied to subobjects within items such as revisions, forms, and datasets which are in most cases always exported with higher-level objects. These options are not available when transferring site ownership.

General options	Description
Include Modified Objects Only	Select this option to include a workspace object only if it was modified since the last time it was exported to the target sites. For example, if only the specification dataset was modified, then it is included and the remaining items are excluded. When exporting to multiple target sites, an object is exported if it was modified since the last export to any site on the list.

General options	Description
Exclude Export Protected Objects	Select this option to exclude workspace objects that are protected through the Access Manager from import/export to remote sites. For example, some of the revisions for an item do not have **Export** and/or **Import** privileges granted at the owning site. When this option is cleared, you receive an error when attempting to import/export the item. By selecting this option, you can import/export those revisions (or other subobjects) that have **Export** and **Import** privileges. **Warning** If you do not know the **Export** and **Import** privileges of a remote item and its subobjects, try to import/export with **Exclude Export-Protected Objects** cleared. If you receive an error message indicating no **Export** or **Import** privilege, then select this option, and try again.
Exclude Folder Contents	Select this option to export only the folder without any of its contents. This is intended for special applications such as exporting part families where family members contained in a folder must be excluded.

Save options	Description
Save All Options As Default	Saves the selected options as the default behavior when importing remote objects.

When importing and exporting datasets, you must decide whether to include all versions and named references associated with that dataset.

Dataset options	Description
Include All Versions	Select this option to include all dataset versions with each dataset selected for import or export. When this option is cleared, the **Include All Versions** option includes only the latest version of each dataset selected for import or export.
Include All Files	Select this option to include all underlying operating system files (such as named references) with each dataset selected for import or export. If you do not select this option, only the dataset metadata is imported or exported. If you import or export a dataset without including the named references, Multi-Site Collaboration automatically retrieves these files from the remote volume into the local FMS cache when they are required. This process improves the performance during the initial import or export, however there is an increase in the time required to open a named reference file for the first time at the remote site.

Dataset options	Description
	Warning When transferring ownership, the **Include All Versions** and **Include All Files** options are automatically selected to ensure that the new owner receives all data with the exported object(s). **Note** For export actions, clearing this option excludes only named reference files. This is to prevent issues with applications which store metadata in **Dataset** objects as named reference forms.

Product structure options	Description
Include Entire BOM	Select this option to include all components if the item selected is an assembly. The revision selectors allow you choose which revision to export with the selected item and its component items, if applicable. You can choose only one revision selector. The **TC_bom_level_export** preference controls whether this option is available.
Transfer Top-Level Item Only	Select this option to transfer site ownership of the selected assembly item and export all components with no site ownership transfer. This option is enabled only if the **Transfer Ownership** option is selected.
Exclude Transfer-Protected Components	When transferring site ownership, select this option to exclude all components that have no **TRANSFER_OUT** and/or **TRANSFER_IN** privileges granted at the owning site. If this option is cleared and a transfer-protected component is found, the import/export operation fails. This option is enabled only if **Transfer Ownership** is selected.
Exclude Export-Protected Components	When exporting with no site ownership transfer, select this option to exclude all components that no export and/or import privileges granted at the owning site. If this option is cleared and a export-protected component is found, the import/export operation fails. **Warning** If you do not know the protection of components at the owning site, try the import/export with the component-related option unset. If you receive an error message indicating lack of privilege on a component, then set the appropriate component-related option, and try the import/export operation again.

Product structure options	Description
Include Distributed Components	Select this option to include components that may be owned by sites other than the site from which you are importing an assembly. Includes distributed components within a distributed assembly. A distributed assembly consists of components owned by more than one site. First, the top-level assembly and all components owned by the assemblies owning site are retrieved. Then individual distributed components are retrieved from their respective owning sites. This option is enabled only when you select the **Include Entire BOM** option. It cannot be used in conjunction with the **Transfer Ownership** option. This option is available only when you select the **Remote Import** option; it cannot be used with an **Interactive Object Export** command. For example, if you are at Site A and are importing an assembly from Site B, that assembly may contain components that are owned not only by Site B but also by Site C and Site D. To import the components owned by Site C and Site D, you must select the **Include Distributed Components** option.

Session options	Description
Preview With Report	If you select this option, no actual import or export object operation is performed. Instead, a dry run of the import or export is performed and a report is generated. During the dry run, all import/export options selected apply. The report contains the list of Teamcenter objects that are exported/imported if the actual operation were to be performed plus the names and size of files. The report also includes error codes and messages for errors that would be encountered during the actual operation. You can print the report or save it to a text or HTML file. For example, if you select the **Include Entire BOM** and **Latest Revision Only** options, the dry run includes the entire product structure using **Latest Revision** as the configuration rule. The dry run also checks the schema between the owning and importing sites and reports any discrepancies and potential problems. This option is enables when you select the **Perform Import/Export In Background** option.

Session options	Description
	This option is mutually exclusive with the **Preview With Report** option.
Continue On Error	Select this option to allow the remote import or export objects operation to continue if errors are encountered while importing/exporting optional objects. The objects that are required are dependent on your data model with many objects related to items being optional. For standard Teamcenter, the following objects attached to an item are required as a minimum: • **Item Revision** • **BOM View** • **BOM View Revision** • **IMAN_master_form** • **IMAN_specification** • **IMAN_requirement** For standard Teamcenter, all relation objects attached to the item revision are considered optional except the following: • **Requirements** • **Specifications** • **Item Master** • **Item Revision Master** If you select the **Continue On Error** option and the **Generate Import/Export Report** option, any error information is included in the import/export report. If you select the **Generate Import/Export Report** option and do not select the **Continue On Error** option, Teamcenter does not generate a report if an error occurs. However, Teamcenter displays an error dialog that identifies the object that caused the error. This option is disabled when you select the **Transfer Ownership** option.

Relationship objects	Advanced tab
Include Reference and Exclude Reference	The include and exclude relations lists are used to define which kinds of related objects are imported and exported. Some relations (for example, **Specifications**, **Requirements**) cannot be excluded; they are essential pieces of the object being imported or exported. However, other relations can be explicitly included or excluded by adding them to the appropriate list using the left and right arrow buttons. When working with change objects, you can add user-defined pseudo folders to change objects, and objects that are placed in these folders have a specific relationship to the change object. You can include or exclude these user-defined relations when importing and exporting change objects. **Note** When exporting a schedule, do not include the **ResourceAssignment** relation. This will cause the export to fail.

Multi-Site Collaboration **Synchronization/Notification** options make it possible to control how the replica is synchronized when the master copy is modified, and whether or not an e-mail notification is received.

Synchronization/ notification options	Description
Synchronize Automatically	Select this option to have the replica automatically synchronized when master data is modified.
Synchronize in Batch Mode	Select this option if you want the replica to be synchronized in batch mode using the sync utility.
Notify By E-mail	Select this option if you want to be notified by e-mail when the master copy is modified.

Change objects options	Description
Include BOMChanges	Select this option to include the **BOMChange** objects associated with the affected assembly of the selected change object during remote export.

Change objects options	Description
Include Supersedures	Select this option to include supersedure objects associated with each **BOMChange** object during a remote export. When you select this option, you must also select the **Include BomChanges** option, because supersedure information can only be transferred within the context of a **BOMChange**. **Note** To include a form associated with the supersedure, the **ECM_Supercedure_to_form_reln** relation must be added to the **Include Reference** list.

Chapter

14 Import and export behavior

Chapter

14 Import and export behavior

The following table describes various Teamcenter objects and their import and export behavior in a Multi-Site Collaboration network.

Teamcenter object	Import export behavior

Archived Objects	Archived objects are exportable, but not transferable. The behavior when opening an archived object is the same at a remote site as it is at the local site. A message displays notifying you that the object is archived. The **Archived Date** property is imported.
	
	You cannot request for a remote object to be restored from a remote site. The following message is displayed:
	
	`Unable to restore object "Object ID"`
	`Object is owned by another site`
	
	Archived objects must be restored at the owning site. Any attempt to transfer ownership of an archived object displays the following message:
	
	`Archived object cannot be exported to another owning site`

Checked-Out Objects	Checked-out objects are exportable, but not transferable. The checkout flag cannot be imported. The flag indicates that someone has the writable instance reserved. At a remote site, the instance is never modifiable.
Objects in Process	Target objects in a release procedure are exportable, but not transferable. The **Process Stage** status, **Audit** file, and **Job** object cannot be imported. You cannot initiate a release procedure on a remote object, nor paste it as a target for release. This also applies to proposing a change using **Change Management** (CM). However, you can paste a remote object as a reference object in a release procedure.
Released Objects	Released objects are transferable. The release status and **Audit** file are exported. Siemens PLM Software recommends that database sites use rules-based object protection to ensure that released objects are protected. Otherwise, when using object-based protection, the released objects inherit the default ACLs of the person performing the remote import.

Teamcenter object	Import export behavior
Bill of Materials (BOM)	Viewing product structure from a remote site requires that the BOM components of the assembly reside in the local database. When importing an item, you have the choice of importing the entire BOM or the top-level only. If only the top level of a BOM is imported, a message is displayed when the BOMView is opened notifying you that the BOM components are not imported. You are asked if you want to import the components. The BOM components can only be imported if they were published. If they were not published, you must either coordinate with the owning site to publish the components, or perform a reimport of the top level assembly using the **Include Entire BOM** option. Remember that the **Include Entire BOM** option imports all levels of the assembly including subassemblies and their component parts.

BOM with Variant Conditions	Viewing variant conditions from a remote site requires that the parent assembly defining the variant rule reside in the local database.
	
	The display of variant conditions displays the following strings to explain why the expression cannot be seen in its entirety. These strings are displayed in lieu of the variant condition:
	
	`<<UNREADABLE OPTION>>`
	`<<REMOTE OPTION`
	`UNCONFIGURED OPTION`
	
	The **Variant Condition** dialog box becomes read-only when opened for such expressions. All buttons except **Cancel** are grayed-out.
	
	Define Defaults and **Variant Rule Check** dialog boxes are not read-only. You cannot modify existing expressions; you can only remove existing expressions or define new expressions.
	
	The **Variant Rule** dialog box shows lines for remote/unreadable options as follows:
	
	`<<XXX OPTION>> in the Option Name column`
	`****** in other columns`
	`You cannot select these lines.`
	
	When evaluating variant conditions, remote/unreadable options are interpreted as undefined (a ? appears in the **Is Configured** column, regardless of the rule).

Teamcenter object	Import export behavior
Requirements objects	When a Requirements object is exported, the associated full-text dataset is exported with it. Therefore, you must select the **Include All Files** check box as a dataset option in the **Remote Export Options** dialog box. If you select the **Export entire BOM** checkbox, all items participating in the BOM View Revision (BVR) are exported. Otherwise, the BVR items are exported as stubs. If you transfer ownership of a Requirements object, by selecting the **Transfer Ownership** check box, the Requirements object at the exporting site becomes a replica and its icon changes to reflect this. You can synchronize replicated Requirements objects, as you do other objects, bye selecting the object and choosing **Multi-Site Collaboration→Synchronization→Object**. You import Requirements objects the same as any other object. Requirement objects can be published to ODS and located using the remote search capability of Multi-Site the same as any other object.

Chapter

15 Remote checkin/checkout

Chapter

15 Remote checkin/checkout

The Remote Checkin/Checkout (CICO) feature is an alternative method to transferring site ownership if you must modify an object that is owned by another site. When to use Remote CICO versus transfer of site ownership is dictated by the type of data you want to change, the nature of the change you want to make and the total size of the data that needs to be transferred. This helps you understand the factors that helps you make the decision as to which method to use. It is not intended to provide the details on how to use this feature.

When to use remote checkin/checkout over transfer of site ownership

Transferring site ownership of an item in order to modify a portion of it can be a time-consuming process because it requires that all revisions and most attachments, including files, be copied to the site that needs to modify the data. For example, you may want to modify an assembly that belongs to another site by adding new components and modifying some of the existing files. To use the transfer site ownership method, you would have to transfer site ownership of the assembly and many of its components. This means transferring site ownership of all the item revisions and most of the attachments including associated files. Getting the data through a WAN is not only time-consuming but can be error-prone because the data transmission is exposed to the risk of network errors for a long period of time.

The remote CICO method avoids the need to transfer site ownership just to gain write access to an object. You only need to replicate the particular object you want to modify, such as a particular item revision or dataset and then gain write access to it by performing a remote checkout operation. The remote checkout operation not only gives you write access to the replica object, but it also prevents other users at other sites from modifying the object before you can complete your changes because a reservation is created on the master copy at the owning site. The reservation not only prevents other users from checking out the master copy, but also from transferring site ownership. This effectively puts a lock on the master copy. After you have completed your changes to the remotely checked out replica, you must perform a remote check in operation which applies your changes to the master copy and releases the lock.

The main advantage of using Remote CICO over site ownership transfer is that the amount of data copied from the owning site is much less. Instead of copying all item revisions and their attachments, only the particular revision you are modifying must be copied. The limitations for remote CICO are you cannot make Variant and ECM changes against a replica.

Considerations

When deploying Multi-Site Collaboration, you must define the use cases that involve modifying a remotely owned object and identify the uses cases where Remote CICO can be used more efficiently than site ownership transfer. Such use cases have the following characteristics:

- The use case falls under the supported use cases described in *Using Multi-Site Collaboration*.
- The nature of the change to be made does not include any of the known limitations.
- The amount of data to be replicated in preparation for Remote CICO is much less than the amount of data to be transferred with site ownership. This is generally true when items have a high number of item revisions and/or large files.

For other use cases, it is necessary to transfer site ownership.

Limitations on remote checkout

There are limitations on what classes of objects can be checked out remotely:

- **BOM view**
- **BOM view revision**
- **Dataset**
- **Folder**
- **Form**
- **Item**
- **Item revision**

Limitations on remote checkin

There are limitations on the classes of objects that an be remotely checked in. These classes cannot be added to a checked out object and later checked in:

- **Engineering change**
- **Incremental change**
- **Variants**
- **Absolute occurrences**
- **Teamcenter's mechatronics process management classes**
- **Manufacturing Process Management classes**
- **Classification data**

In general, only generic classes such as items, revisions, datasets, and forms can be added to a checked out object.

Operations on remotely checked out objects

You can perform the following operations on remotely checked out objects:

- Add BOM view and BOM view revisions
- Add release status
- Add/remove attachments
- Add/remove assembly components

- Modify dataset file
- Modify form attributes
- Modify properties
- Revise an item revision

Note

When you revise a remotely checked out item revision, the attached dataset files are copied to the new revision only if they are not checked out to the remote site.

Remote CICO of sequences

A sequence represents a complete iteration within an item revision. You can check out only the latest sequence. For remote checkout, the item revision is checked out of the owning site, exported to the remote site, and checked out locally at the remote site. For remote checkin, changes to the item revision is imported to the owning site, it is checked in to the owning site, and then it is checked in locally at the remote site. Canceling a remote checkout for an item revision sequence discards any changes made to the item revision and makes it available for modification by other users.

For information about sequence checkin and checkout, see the *Rich Client Interface Guide*.

Working with remote arrangements

Multi-Site Collaboration supports NX arrangements that are relationships to assemblies or a BOM view revision (BVR). The following relations must be included when you do a remote check out and check in of an NX assembly that contains arrangements:

- **TC_Arrangement**
- **TC_DefaultArrangement**
- **TC_BaseArrangementAnchor**

Your administrator can configure Multi-Site Collaboration to include these relationships automatically.

For more information, see *Remote checkin and checkout administration*.

Remote CICO and data_share utility

The **data_share** command line utility implements some features that are intended to help both the end user and the system administrator deal with various Remote CICO-related situations. The following options are available in **data_share** utility:

Option	Description
list_remote_co	Use this at the owning site to list the objects that are checked out by remote users.

Option	Description
list_replica_co	Use this at the replica site to list the replica objects that are checked out from their respective owning sites.
cancel_remote_co	Use this at the owning site to force the cancellation of checkouts of locally-owned objects by remote users. Note that this attempts to also cancel the replica checkout at the remote site. If the cancellation of the master copy checkout succeeds but the cancellation of the replica checkout fails, make sure to inform the remote user so that administrator can manually cancel the replica checkout.
cancel_replica_co	Use this at the replica site to cancel the checkout on a replica. Use this only if it is known that the replica is checked out but the master copy is not checked out. To cancel a checkout where both the replica and master copy are checked out, use the **Portal Tools→Checkin/Checkout→Cancel Checkout** menu option.

The **data_share** utility help (**-h** argument) provides additional information.

Error recovery procedures

For various reasons, it is possible that the master copy and the replica checkouts could end up in an inconsistent state wherein one is checked out but the other is not. This is an error condition that needs to be corrected manually.

If the master copy is checked out but the replica is not, this condition is most likely caused by a failure in the middle of a remote checkin process. Use the **data_share** utility at the owning site to cancel the remote checkout using the **-f=cancel_remote_co** option. After this, you log on to the replica site and perform a remote checkout on the replica containing the changes were made to it and check the object back in.

If the replica is checked out but the master copy is not, this condition is most likely caused by someone at the owning site forcibly canceling the remote checkout on the master copy. You must use the **data_share** utility at the replica side to cancel the replica check out using the **-f=cancel_replica_co** option. After this, you can log on to the replica side using the rich client and perform a remote checkout of the replica, which should still have the changes that were made to it, and check the object back in.

Chapter

16 Importing remote objects

Chapter

16 Importing remote objects

If your site has a network connection to a remote site, you can use Multi-Site Collaboration to import objects from other sites.

To import a remote object using Multi-Site Collaboration, the object must first be published into an Object Directory Services (ODS) site by the site that owns the object. You must then search the ODS site for the specific objects you want to import using the find remote application. Once you have found the objects, you can use the **Import→Remote** commands on the **Tools** menu. A series of dialog boxes enable you to import the remote objects into your local database. The **Remote Import Progress** dialog box displays the object name, operation, and status of both active and completed remote import operations until it is closed. Once closed, completed operations are no longer displayed.

After the remote import operation completes, your database contains a read-only replica of the objects that were imported.

Preferences

Before remote import operations are performed, you can specify options for what you want to import and how it is to be imported. For example, you can specify a revision selector when importing an item so that only a specific revision, such as the latest released revision, is imported. You can also specify whether to include components when importing an assembly. The system displays the **Import Remote Option Settings** dialog box when you click **Import Remote Option Settings**  in the **Import Remote Options** dialog box.

Note

By default, objects related by the following reference relationships are imported along with selected objects:

- **TC_ic_intent_rtype**
- **IMAN_master_form**,
- **IMAN_requirement**
- **IMAN_specification**.

The references included for import are displayed on the **Advanced** tabbed page of the **Import Remote Options** dialog box.

The references displayed in this list are determined by the values of the **TC_relation_required_on_transfer** and **TC_relation_required_on_export** preferences.

Remote import and transfer of ownership

You cannot modify replica objects in your database. Therefore, to modify an object from a remote site you must transfer ownership of the object to your site. Before you can transfer site ownership, the site that currently owns the object must grant your site the **TRANSFER_OUT** privilege for the object.

To transfer site ownership, you must set the **Transfer Ownership** option in the **Import Remote Options** dialog box. When you set this option, several options in the dialog box are automatically disabled and other options are automatically set. For example, you can no longer specify an **Item Revision Selector**, and the **All Revisions** option is automatically set. This is because when transferring site ownership of an item, you must take ownership of all revisions of the item.

After a successful transfer of ownership, the original master copy becomes a read-only replica. The object in your database becomes the master copy and you can now modify the object.

The following table describes the import/export behavior of data objects in various states within a Multi-Site Collaboration network.

Object	Behavior description
Checked-out objects	Checked-out objects are exportable, but not transferable. The Check-Out flag cannot be imported. The flag is an indicator that someone has reserved the writable instance. At a remote site, the instance is never modifiable. An attempt to transfer ownership on a checked-out object will display the following message: `Checked-out object cannot be exported to another owning site` This protects objects that are exclusively reserved or are actively being modified by another user. The checked-out object cannot be transferred until the object is checked in.
Objects in workflow jobs	Target objects in a release procedure are exportable but not transferable. The Process Stage status, Audit file, and Job objects cannot be imported. You cannot initiate a release procedure on a remote object, nor paste it as a target for release. This also applies to proposing a change using Change Management (CM). However, you can paste a remote object as a reference object in a release procedure. The following message is displayed when attempting to release a remote object: `Object ID is a read only copy`

Object	Behavior description
Released objects	Released objects are transferable. The release status and audit file are exported. Siemens PLM Software recommends that database sites use rules-based object protection to ensure that released objects are protected. Otherwise, when using object-based protection the released objects inherit the default ACLs of the person performing the remote import operation.
Bills of Materials (BOM)	Viewing product structure from a remote site requires that the BOM components reside in the local database. When importing an item, you have the choice of importing the entire BOM or only the top-level item. If only the top-level item is imported, a message is displayed when the BOM view is opened notifying you that the BOM components have not been imported. You are asked if you want to import the components. The BOM components can only be imported if they were published. If they were not published, you must either coordinate with the owning site to publish the components or perform a reimport of the top level assembly using the **Include Entire BOM** option. Remember that this option imports all levels of the assembly, including sub-assemblies and their component parts.

BOM with variant conditions	Viewing variant conditions from a remote site requires that the parent assembly that defines the variant rule must reside in the local database. The display of variant conditions displays the following strings to explain why the expression cannot be seen in its entirety. These strings are displayed in lieu of the variant condition:
	`<<UNREADABLE OPTION="">>`
	`<<REMOTE OPTION="">>`
	`<<UNCONFIGURED OPTION="">>`
	The **Variant Condition** dialog box is read-only when opened for such expressions. All buttons except **Cancel** are disabled. The **Define Defaults** and **Variant Rule Check** dialog boxes are not read only. You cannot modify existing expressions, you can only remove existing or define new expressions. The **Variant Rule** dialog box shows lines for remote/unreadable options as follows:
	`<<XXX OPTION>>`
	in the **Option Name** column
	****** in the other columns
	You cannot select these lines.
	When evaluating variant conditions, remote/unreadable options are interpreted as undefined (a question mark (**?**) appears in the **Is Configured** column, regardless of the rule).

Object	Behavior description
Objects in projects	When objects in projects are exported, the explicitly assigned project IDs are exported with the other object data. When an object in a project is imported, it is assigned to the project that has the same project ID as the imported object. If an imported object has multiple project IDs, the object is assigned to all of the applicable projects that can be located on import. New projects will not be created if a match is not found. The ID matching is performed in a case-sensitive manner; therefore, project IDs must exactly match at both sites in order to assign imported objects to a project. When an imported object (replica) is assigned to an ID-matched project, the project propagation rules at the import site are invoked to assign attached objects to the project. Siemens PLM Software recommends that projects be duplicated across sites before attempting to share project data.
Requirements objects	When a Requirements object is exported, the associated full-text dataset is exported with it. Therefore, you must select the **Include All Files** check box as a dataset option in the **Remote Export Options** dialog box. If you select the **Export entire BOM** checkbox, all items participating in the BOM View Revision (BVR) are exported. Otherwise, the BVR items are exported as stubs. If you transfer ownership of a Requirements object, by selecting the **Transfer Ownership** check box, the Requirements object at the exporting site becomes a replica and its icon changes to reflect this. You can synchronize replicated Requirements objects, as you do other objects, bye selecting the object and choosing **Multi-Site Collaboration→Synchronization→Object**. You import Requirements objects the same as any other object. Requirement objects can be published to ODS and located using the remote search capability of Multi-Site the same as any other object.

Import remote objects

1. Choose **Tools→Import→Remote**.

   The system displays the **Import Remote** dialog box.

2. Type the reason for importing the remote object in the **Reason** box.

3. Click **Import Remote Option Settings**  to access the **Import Remote Options** dialog box.

4. Click **Yes** to start the remote import operation.

   The **Import Remote Options Setting** confirmation dialog box displays with the current remote option settings.

   - Click **Yes** to complete the remote import operation.

   - Click **No** to return to the **Import Remote Options** dialog box, then click **No** in the **Import Remote Options** dialog box to cancel the operation.

   The system displays the **Remote Import Progress** dialog box. The object name, operation, and progress status of both active and completed remote import operations are displayed in the dialog box until it is closed. Once closed, completed operations are no longer displayed.

Chapter

17 Modifying remote objects

Chapter

17 Modifying remote objects

Multi-Site Collaboration provides two methods for sharing write access to shared data when a remote site needs to modify data currently owned by another site:

Method	Description
Transferring Site Ownership	Transfer ownership of an item, modify the item, then transfer site ownership back to the original owning site. Requires transferring site ownership of all revisions and most attachments and files. If you transfer an item revision with a sequence, its sequence manager is also transferred.
Remote CheckIn and Checkout	Replicate item, then check out only the objects requiring modification. Only the latest sequence can be checked out remotely. A replica dataset with deferred files can be remote checked out to gain write access. Opening the dataset retrieves the file into the local FMS cache. Modifying the file and saving it creates a new version of the dataset with a local **ImanFile** name reference and volume file. Upon remote checkin, these new dataset versions, **ImanFile** objects, and volume files are transferred back to the owning site. If the remote checkin is successful, the new **ImanFile** objects are marked as deferred and the related volume files deleted at the replica site. When a replica is checked out, a remote checkout is performed at the item's owning site, ensuring no other user in the network can modify it.

Using remote checkin and checkout

The method of transferring site ownership is time consuming for large items. You can use emote checkin and checkout modify a remotely owned object.

Use the remote checkin/checkout functionality to modify remote objects in the following circumstances:

- Modifying attachments
- Adding and removing attachments
- Adding a new item revision

- Adding components to an assembly
- Adding components to an item containing an existing BOM view but no BOM view revision
- Adding components to an item revision with an existing BOM view revision
- Adding components using Teamcenter Integration for NX

The following examples illustrate use cases for these circumstances.

Modify attachments

In the following use case, a remote user modifies a specification dataset of a remote item revision:

Step	Remote site	Owning site
1	User replicates the item, item revision, and the desired attachment.	Requested objects are exported and sent to requesting site.
2	User checks out the specification dataset. Because the dataset is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica dataset ensuring no other user at this site can modify it.	The IDSM checks out the dataset on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the dataset. The item site ownership cannot be transferred.
3	If the user decides to cancel the checkout, a remote checkout cancellation is sent to the owning site and the local checkout is also canceled.	The IDSM cancels the checkout on behalf of the remote user.
4	User modifies the dataset.	
5	User checks in the modified dataset. The updated dataset is exported from the database and sent to the owning site. After a successful remote checkin, a local checkin is performed.	The updated dataset is imported and the IDSM performs a check in. The updated dataset is now available for other users to check out.
Notes:	To modify other attachments before the first dataset is checked in, repeat steps 1 and 2 for the desired attachment.	While an attachment is checked out, it can be replicated by other sites but not by the site that checked it out.

Note

The IDSM user must have write access to a master item at the owning site to make changes to remote item replicas. You must make the IDSM user a member of the dba group or change the rule tree to grant write access. The replica revision fails with the error: **No Write access to master item**.

Add a new item revision

In the following use case, a user adds a new item revision to an existing remote item:

Step	Remote site	Owning site
1	User replicates the item and at least one item revision.	Requested objects are exported and sent to requesting site.
2	User performs a remote check out on the item and then performs a revise action on the item revision, completing the necessary information. Because the item revision is a replica, the revise request is sent to the owning site.	The IDSM creates the requested item revision.
2a	After receiving a successful status for the item revision creation, a remote import request for the new item revision is automatically issued.	The IDSM exports the new item revision and sends it to the remote site.
2b	Upon successful import of the new item revision, the window is refreshed to show the new item revision.	
3	User can perform a remote checkout of the item revision and make the modifications on the newly created item revision.	
Notes:	If the user clicks the **Assign** button in the **Save As** dialog box, the assign function is sent to the owning site, which assigns the new item revision ID. While the new item revision is being created at the owning site, the replica item is locked to prevent other users at this site from performing the same operation.	A means to integrate custom part numbering schemes needs to be implemented.

Note

The IDSM user must have **Write** access to a master item at the owning site to make changes to remote item replicas. Otherwise, the replica revision fails and returns an error. To avoid this error, you must make the IDSM user a member of the **dba** group or change the rule tree to grant the user write access. For example, if the IDSM process is run by the **idsmuser** user, use Access Manager (AM) to modify the **Import/Export** rule for the **idsmuser** user to allow **Write** access.

For information about modifying rules, see the *Access Manager Guide*.

Add components to an item with no existing BOM view

In the following use case, a user adds components to an existing item revision owned by another site. The item revision does not contain a BOM view. The checkout is performed from My Teamcenter:

Step	Remote site	Owning site
1	User replicates the item and the item revision to which components will be added. User expands the item and sees no BOM view exists.	Requested objects are exported and sent to requesting site.
2	User checks out the item. Because the item is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica item, ensuring no other user at this site can check out the replica item.	The IDSM checks out the item on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the item. Also, the site ownership of the item cannot be transferred.
3	User selects the item revision and performs a checkout. Because the item revision is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica item revision, thereby ensuring that no other user at this site can check out the replica item revision.	The IDSM checks out the item revision on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the item revision.
4	User sends the item revision to Structure Manager and adds the necessary components. Structure Manager displays a message stating that no structure data exists and gives the user the option to create them.	

Step	Remote site	Owning site
5	After all components are added, the user exits Structure Manager and checks in the item revision. The item, item revision, BOM view, and BOM view revision are exported and sent to the item's owning site (with transfer of ownership for the BOM view and BOM view revision). All occurrences are stubbed.	The IDSM imports the objects and then checks in the item revision.
6	User checks in the item. A checkin request is sent to owning site. After successful remote checkin, local checkin is performed.	The IDSM checks in the item.

Add components to an item containing an existing BOM view but no BOM view revision

In the following use case, a user adds components to an existing item revision owned by another site. The item revision contains a BOM view, but no BOM view revision. The checkout is performed from My Teamcenter:

Step	Remote site	Owning site
1	User replicates the item and the item revision to which components will be added. User expands the item and sees that a BOM view exists. User expands the item revision and sees that no BOM view revision exists.	Requested objects are exported and sent to requesting site.
2	User checks out the item revision. Because the item revision is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica item revision ensuring no other user at this site can check out the replica item revision.	The IDSM checks out the item revision on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the item revision.
3	User sends the item revision to Structure Manager and adds the necessary components. Structure Manager displays a message stating that no structure data exists and gives the user the option to create them.	

Step	Remote site	Owning site
4	After all components are added, the user exits Structure Manager and checks in the item revision. The item, item revision, and BOM view revision are exported and sent to the item's owning site (with transfer of ownership for the BOM view revision). All occurrences are stubbed.	The IDSM imports the objects and then checks in the item revision.

Add components to an item revision with an existing BOM view revision

In the following use case, a user adds components to an existing item revision owned by another site. The item revision contains a BOM view revision. The checkout is performed from My Teamcenter:

Step	Remote site	Owning site
1	User replicates the item and the item revision to which components will be added. User expands the item and sees that a BOM view exists. User expands the item revision and sees that a BOM view revision exists.	Requested objects are exported and sent to requesting site.
2	User checks out the item revision. Because the item revision is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica item revision, ensuring no other user at this site can check out the replica item revision.	The IDSM checks out the item on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the item.
3	User checks out the BOM view revision. Because the BOM view revision is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica BOM view revision, ensuring no other user at this site can check out the replica BOM view revision.	The IDSM checks out the BOM view revision on behalf of the remote user.
4	User sends the item revision to Structure Manager and adds the necessary components.	

Step	Remote site	Owning site
5	After all components are added, the user exits Structure Manager and checks in the BOM view revision. The BOM view revision is exported and sent to the item's owning site. All occurrences are stubbed.	The IDSM imports the objects and then checks in the item revision. **Note** In a four-tier environment, if you have previously performed a remote import action with the **Include Entire BOM** option selected, the occurrences are owned by the BOM view revision's owning site instead being stubbed. This behavior continues until you exit the Teamcenter client.
6	User checks in the item.	The IDSM checks in the item.

Add components using Teamcenter Integration for NX

In the following use case, a user adds a component to a remote item revision using NX. Before running NX, the user must check out the **UGMASTER** dataset and the item revision BOM view revision using the **Check-in/Check-out** command on the rich client **Tools** menu.

Note

Teamcenter Integration for NX always attempts an implicit checkout using a special reservation type that does not trigger a remote checkout. Thus, users must perform an explicit remote checkout to modify a remotely owned object with Teamcenter Integration for NX. Using this method, all modifications are sent to the owning site after an explicit checkin of the object. This would not occur using the implicit checkin Teamcenter Integration for NX routinely performs.

Step	Remote site	Owning site
1	User replicates the item, item revision and the **UGMASTER** dataset of the desired item revision.	Requested objects are exported and sent to requesting site.
2	User checks out the replica **UGMASTER** dataset. Because the dataset is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica dataset ensuring no other user at this site can modify the replica dataset.	The IDSM checks out the dataset on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the dataset. The item's site ownership cannot be transferred.

Step	Remote site	Owning site
3	User checks out the item revision's BOM view revision. Because the BOM view revision is a replica, the checkout request is sent to the owning site. If the remote checkout request is successful, a local checkout is also performed on the replica BOM view revision ensuring no other user at this site can modify the replica BOM view revision	The IDSM checks out the BOM view revision on behalf of the remote user. At this point, no other user (either remote or local to the owning site) can check out the BOM view revision. The item's site ownership cannot be transferred.
4	Using NX, user retrieves the assembly. With automatic locking on, NX initiates implicit check out requests for the **UGMASTER** dataset and the BOM view revision. The system checks if the objects are already checked out by the user and if so, allows NX to modify the replica.	
5	User adds components to the assembly. User may save the changes and exit NX, then continue with additional changes at a later time. No remote checkin is performed during the save operations.	
6	After all changes are made, user checks in the modified **UGMASTER** dataset and the BOM view revision using the rich client interface. The updated objects are exported from the database and sent to the owning site. After a successful remote check in, a local check in is performed.	The updated objects are imported and the IDSM performs a checkin. The updated object is now available for other users to check out.

Note

When you create a new dataset under an existing item revision, the site owning the item revision owns the new dataset. Original ownership/creation data is not tracked.

Automatic remote checkin and checkout for baseline functionality

Baselines are snapshots of working item revisions and assembly structures that provide historical pictures of product items. These snapshots allow you to freeze data at a particular stage and share that data. Baselines help you to:

- Share their WIP designs to suppliers/OEMs.
- Capture alternative designs created during the product development.
- Preserve their WIP data for historical purposes.

Baselines do not contain the attachments that are excluded during remote import at the replica site.

Baseline are an image of the item revision as seen at the remote site and not at the owning site, that is, if there are some attachments that are excluded during the remote import at the replica site, the baseline does not contain these attachments, irrespective of the business rules that are specified.

Baseline revision ID is unique

When a replicated item revision is baselined, it is required that the baseline revision ID that is created is unique. The baseline revision ID is verified at the owning site.

Baseline follows business rules of the owning/replica site

Baseline follows the business rules of the same (owning/replica) site as that followed by revise/save as to avoid inconsistencies. This is not possible in the following cases:

- If the structure has components from many sites, the baseline must be created only at the owning site, due to excessive network traffic.
- If rollback of baseline is needed. Deletion of objects in an RPC call is not supported, so rollback of baselines is not possible.
- When information is needed about the baseline workflow templates at the owning site and at the replica site.

The owning site cannot create a baseline when the item is checked out by a remote replica site. When an item is remotely checked out at a replica site, the owning site or any other replica site cannot modify the item. Two sites cannot have **CO** (checked out) status of a single object. Only a single site can have **CO** status of an object.

Baseline should be owned by the owning site after checking in the item. Until the item is checked In, the baseline that is created is owned by the replica site. After performing the remote check In, the baseline is owned by the owning site.

Baseline of replicated structures

Replicated structure baselines must follow essentially the same rules as local structures. The baseline revision ID for the top line is generated locally. The ID is verified at its owning site, where the remote child components are generated at the owning site.

Preference to perform baseline of replicated objects

The **Baseline_auto_remote_checkout_allowed** site preference allows you to perform auto remote checkout. Its value can be set to **ON/OFF**. If the preference is turned **OFF**, then you perform the remote checkout manually or do a remote import with transfer ownership to get write access to **Items/BOM views**. With the

preference value turned set to **ON**, the replica **Items/BOM views** is checked out automatically during baseline operation. For more information on this preference see the *Preferences and Environment Variables Reference*.

Autoremote checkout

In case of a piece part, baseline creation needs write access to the item. For a structure, baseline requires write access to the item as well as the BOM view. Baseline performs the remote checkout of the item/BOM view, when it baselines a remote component.

Remote checkin

When the baseline operation is complete, it must check in the objects that were explicitly checked out.

Remote cancel checkout

If you encounter errors during the baseline creation, you may rollback using **Remote Cancel Check Out** on the objects that were **Remote Checked Out**.

Chapter

18 Sharing data with unconnected sites

Chapter

18 Sharing data with unconnected sites

If your Multi-Site Collaboration network includes sites that are not connected to through LAN or WAN, use the basic import/export procedures described in the My Teamcenter help to share information with these sites. Be sure to physically transfer any exported objects to the destination site using File Transfer Protocol (FTP) or removable media following the object export operation at your site.

Chapter

19 Updating an object or BOM

Chapter

19 Updating an object or BOM

There are two reasons for updating a remote object:

- To retrieve the latest information.

  Updates ensure that the replicated objects in your local database are completely current.

- To retrieve additional information.

  Updates expand the amount of local data (for example, the entire BOM, all dataset versions) you are storing for a replicated object.

Although you can always obtain more information for a remote object, you cannot reduce the amount of information you are storing for a remote object. Therefore, it is always best to import the least amount of data and add more later.

Chapter

20 Update a remote object

Chapter

20 Update a remote object

Updating a remote object is substantially the same as importing a remote object, except instead of performing a remote search to locate a remote object, you select a remote Teamcenter object and import it.

1. Select a remote object in the rich client.
2. Choose **Tools→Import→Remote**.

   Note

   Steps 3 through 5 are optional. Perform these steps only if you want to use non-default import options. Otherwise, go to step 6.

3. In the **Remote Object Import** dialog box, click **Options**.
4. In the **Import Remote Options** dialog box , set your import options.

   Note

   Unless you want to use other option settings, Siemens PLM Software recommends that you use the following default settings:

   Transfer Ownership = unset

   Include Entire BOM = unset

   Include All Versions = unset

   Include All Files = set

5. Click **OK** to save the settings.

   The **Import/Export Options** dialog box closes.

6. In the **Remote Object Import** dialog box, click **OK**.

Chapter

21 Update a remote BOM

Chapter

21 Update a remote BOM

To view product structure, all bill of materials (BOM) components for the entire assembly must be stored at your site. If an assembly has been imported from another site without setting the **Include Entire BOM** option, it is incomplete and cannot open. You must perform an update of that BOM view object to view it.

1. Double-click a remote BOM view object in the rich client.

   If this BOM is incomplete, the following message appears:

   ```
   The BOM view contains remote items
      Do you want to import them?
   ```

   Note

   If this BOM view opens, it is complete and you do not have to perform this procedure.

2. Click **Yes**.

   The **Objects** dialog box appears listing all components used in this assembly.

3. Select all components in the list and click **Import**.

4. In the **Remote Object Import** dialog box, click **Apply to All**.

   The **Remote Object Import** dialog box closes.

Chapter

22 Using remote inboxes

Chapter

22 Using remote inboxes

You can subscribe to inboxes at remote locations and manage tasks in those inboxes, if you have the appropriate privileges. Remote inboxes allow you to interact with workflow tasks that originated at a remote site. Similar to local inboxes, remote inboxes contain **Tasks to Perform** and **Tasks to Track** folders. However, unlike local inboxes, remote inboxes cannot be expanded in the tree display. You must click the link to the **Inbox** and Teamcenter launches a separate session displaying the remote inbox.

Subscribe to a remote inbox

From My Teamcenter, you can subscribe to remote inboxes (inboxes at sites other than your home site), as follows:

1. Choose **Tools→Remote Inbox Subscription**.

   Any remote inboxes to which you are subscribed are listed as **Selected Inboxes**; any remote inboxes to which you are not subscribed are listed as **Available Inboxes**.

2. To subscribe to one or more of the available inboxes, select each inbox in the **Available Inboxes** list and click the add (**+**) button. To unsubscribe from any of your subscribed inboxes, select the relevant inboxes in the **Selected Inboxes** list and click the remove (**–**) button.

3. When the subscriptions are listed correctly, click **OK** or **Apply**.

   Teamcenter displays the **Subscribe Remote Inbox** dialog box. This dialog box shows the progress of each subscription request.

Working with the data associated with tasks in your remote inboxes

There are two methods of gaining access to the target data of tasks in your remote inboxes; *remote checkout* and *remote export*. The **Remote Checkout** command allows you to access modifiable replicas of the target data associated with the tasks assigned to you. The **Remote Export** command allows you to access read-only replicas of data or transfer site ownership of the data required to perform your tasks.

Check out an object from a remote site

1. Select the object and choose **Tools→Multi-Site Collaboration→Send→Remote Checkout**.

2. In the **Remote Checkout** dialog box:

   a. Type the change number associated with the checkout request in the **Change ID** box.

   b. Type the reason for the checkout request **Comments** box.

   c. Select the site where the object should be sent in the **Target Site** list.

      Click the **Home** button to the right of the **Target Site** field to choose sites from the list.

   d. **OK to remote checkout?**

      Displays the status of objects being remotely checked out.

3. Click **Yes**.

   The system displays the current options in the **Remote Checkout Options Settings** dialog box.

4. Click **Yes** to continue or **No** to cancel the checkout operation.

   The system displays the status of the checkout operation.

Export data to a remote site

1. From My Teamcenter, select the object and choose the **Tools→Multi-Site Collaboration→Send→Remote Export**.

2. In the **Remote Export** dialog box:

   a. Type the reason for the data export in the **Reason** box.

   b. Select the sites to send the objects to in the **Target Sites** list.

      Click the **Home** button to the right of the **Target Site** field to choose sites from the list.

   c. **OK to remote export?**

      Displays the status of objects being remotely exported.

3. Click **Yes**.

   The system displays the current options in the **Remote Export Options Settings** dialog box.

4. Click **Yes** to continue or **No** to cancel the export operation.

   The system displays the status of the export operation.

Chapter

23 Data replication

Chapter

23 Data replication

Data replication through Teamcenter import and export functions is the foundation of Multi-Site Collaboration. In contrast, most other distributed solutions simply export a copy of an object into a remote application's memory and either discard it when the application exits or save it by reimporting the new version back in to the original database.

The latter approach has the advantage of using less disk space because each object has only one disk copy in the entire network. However, it results in poor performance, because the object must be transmitted over the network each time a remote user wants to access it.

The data replication approach used by Multi-Site Collaboration uses more disk space because objects are replicated at various sites. However, once an object has been *copied* to another site, accessing is as fast as accessing any object in the local database.

A replication-based distributed solution must address the following considerations:

- Data integrity

  As an object is replicated to various sites, how do you determine which copy is the latest version of an object? This is especially true if users are allowed to modify replicated objects.

- Security

  Without proper security controls, replicated product information could fall in to the hands of people not authorized to have it.

- Auditing and tracking

  A replication-based system must provide some method of tracking all replicas of an object not just for audit purposes, but also for ensuring that all replicas are updated when the original is modified.

- Rules

  Multi-Site Collaboration addresses these considerations by imposing the following rules on object replication:

  – Only the master object can be replicated.

    You cannot replicate replica objects. When an object is initially created and saved in a database, that instance is considered the master object until such time as it is exported with transfer of ownership.

  – Only the master object can be modified.

All replicas of the master object are read-only. This ensures that the master object is always the latest copy.

– When you export an object, you must specify which sites are authorized to import it.

  This ensures that no unauthorized replicas are made and stores tracking information with the master object.

– When transferring ownership to another site, only one site can be specified.

  This ensures that there is only one master object in the network.

– After it has been replicated, a master object cannot be deleted until all replicas have been deleted.

  These rules ensure network-wide referential integrity.

Chapter

24 Using synchronization

Chapter

24 Using synchronization

You can synchronize data between sites manually or automatically. Teamcenter also provides synchronization utilities, however to use them you must have administrator privileges.

Data synchronization

When a master object is replicated at other sites, it is necessary to update the replicas whenever the master object is modified. The process of updating replicas is referred to as *synchronization*.

There are important factors that your site administrators must consider when planning data synchronization.

Data synchronization options

Synchronization options are set in the **Import Remote Options** dialog box. You can choose between *automatic* synchronization and *batch* synchronization. You can also choose to be notified when the master object is modified.

Choose automatic synchronization when you have imported a replicated object and want to specify that your replica is to be synchronized immediately after the master object is modified. This results in an efficient and evenly distributed synchronization process in which replicas are updated minutes after the master copy is modified.

Additionally, you can request to be notified when the master object of your replica is modified by selecting the **Notify by E-mail** option.

For complete option descriptions and requirements, see *Data synchronization options*.

Note

Auto synchronization can only be used when importing remote objects; it cannot be used when performing interactive object export.

Choose batch synchronization when you have imported a replicated object and want the administrator at the owning site to synchronize your replica with the master object. The synchronization is performed using the **data_sync** utility; your replica and any other replicas defined for the utility are synchronized in a single batch. When you choose this method, the synchronization is performed at the time scheduled by the owning site administrator, rather than immediately as with automatic synchronization.

Define a synchronization method

1. Select the imported replica to be synchronized.
2. Click **Tools→Import→Remote**.
3. In the **Import Remote** dialog box, click **Import Remote Options** in the lower right corner.
4. In the **Import Remote Options** dialog box, click the **Advanced** tab.
5. In the **Synchronization/Notification Options** pane, select the type of synchronization and/or notification required. For option descriptions and requirements, see *Data synchronization options*.

Default synchronization behavior

If none of the options are set in the **Import Remote Options** dialog box, the default synchronization behavior for imported replicas is as follows:

- If the object is being imported for the first time, the default synchronization method is through batch mode using the **data_sync** utility. There is no notification
- If the object was previously imported, the option settings that were last set are used.

Visualization data synchronization

In a Multi-Site Collaboration environment, you can develop components and sub-assemblies at multiple sites while the entire assembly is configured at a single site. For collaborative design tasks, such as design reviews, you view visualization data for your parts, components, and assemblies. For the collaborative design tasks to be effective, the visualization data created during the collaborative tasks must be replicated and synchronized, along with the original (derived) visualization data. Because the visualization data is usually for a sub-assembly or assembly, Multi-Site Collaboration manages direct model (JT) files, 2D drawings, images, and documents associated to an item revision. Authored visualization data is data that references the derived visualization data in order to create higher level visualization functionality, and this data is authored directly by the core visualization tools. Examples of authored visualization data include PLM XML structure captures, markups, product views, sessions, and work instructions. When you create a visualization session or mark ups of an existing visualization session, the visualization data is replicated and synchronized to the sites where the original (derived) visualization is replicated.

The synchronization includes visualization datasets directly related to item revisions, and visualization datasets related to datasets related to item revisions. If a visualization dataset is related to a CAD dataset (and not the item revision) and the site intent is visualization synchronization, this visualization dataset is not replicated and synchronized because its parent item is not replicated and synchronized.

Synchronization on-demand

The rich client provides **Tools** menu commands that allow you to obtain the synchronization state, or synchronize a specific component or assembly. There is also a command that allows you to obtain the synchronization state of an object. All of these commands are also available from a shortcut menu. Synchronization state indicates whether the replica object is up to date or indicates whether an object that has been added to the master has been replicated by the site where you perform the synchronization.

Component synchronization allows you to determine the synchronization state of a specific component revision and all objects associated with it, such as BVR and attachments. If objects associated with the component are out of date, you can initiate synchronization and visually verify whether the synchronization succeeded or failed. You can also initiate synchronization of a selected component directly without first determining its state.

Object-level synchronization allows you to determine the synchronization state of individual objects, such as a dataset or form. Items and item revisions can be selected for object-level synchronization, but in this case, the synchronization state of their associated objects is not displayed.

Assembly synchronization allows you to determine the synchronization state for the components in an assembly of a specific BOM revision. Each component in the assembly is traversed until an out-of-date component is found or a leaf node is detected. If a component is out of date, you can initiate synchronization and visually verify whether the synchronization succeeded or failed. You may also initiate synchronization directly, which causes the assembly components to be synchronized, as required, to bring the assembly up to date without first determining the component state.

You set the preferences for on-demand synchronization of assemblies and components in the **Synchronization Preferences** dialog box. This dialog box is accessed from a button on the dialog box that appears after you select the synchronize command.

Select this	To
Report Only	Generate a synchronization state report for the selected object, assembly, or component.
Perform Sync	Synchronize the replicas of the selected assembly or component. To verify the results of the synchronization, you must synchronize again in report only mode.
Perform Sync in Background	Synchronize the replicas of the selected assembly or component in a background process that allows you to continue working while the synchronization process completes. Teamcenter displays a **Sync Progress** dialog box that shows the progress of the process. To verify the results of the synchronization, you must synchronize again in report only mode.

Select this	To
Specific Revision Rule	Select the revision rule to use for the report or synchronization from a list of local site revision rules. When synchronizing item revisions, **Selected Revision** appears as the value for the revision rule. This indicates that the selected revision is the configured revision to be synchronized. By default, the revision rule list contains rules define at the local site. However, this can be overridden by the **TC_sync_revision_rules** site preference. **Note** If you are synchronizing or getting a report for the synchronization state of a item, you must select the revision rule to identify the specific revision you desire.
Exclude Folder Contents	Export only the folder without any of its contents. This is intended for special applications such as exporting part families where family members contained in a folder must be excluded.
Exclude Export Protected Objects	Exclude workspace objects that are protected through Access Manager from import/export to remote sites. For example, some of the revisions for an item do not have export or import privileges granted at the owning site. When this option is not set, you receive an error when attempting to import or export the item. By setting this option, you can import or export those revisions (or other subobjects) that have export and import privileges.
Save All Options as Default	Save the selected options as the default settings.
Include Entire BOM (Available only for assemblies.)	Include all BOM components. This option is display only and is always selected. The revision rule allows you choose which revision to export with the selected item and its component items, if applicable.
Exclude Export Protected Components (Available only for assemblies.)	Exclude all components that do not have export or import privileges granted at the owning site. If this option is not set and an export-protected component is found, the import/export operation fails.
Generate Failure Report	Generate a report showing errors that occurred during the synchronize process. This option is available for **Perform Sync** and **Perform Sync in Background** only.

The **Advanced** tab provides relationship options that allow you to exclude or include specific types of related objects from the synchronization. The **Include Reference** and **Exclude Reference** lists are used to define which kinds of related objects are

imported and exported. Some relations (for example, Specifications, Requirements) cannot be excluded – they are essential pieces of the object being imported or exported. However, other relations can be explicitly included or excluded by adding them to the appropriate list using the left and right arrow buttons. When working with change objects, user-defined pseudo folders can be added to change objects, and objects that are placed in these folders have a specific relationship to the change object. These user-defined relations can also be included or excluded when importing and exporting change objects.

The remote import performed during the synchronize process always includes a workspace object only if it was modified since the last time it was exported to the target sites. For example, if only the specification dataset was modified, then it is included and the remaining items are excluded. When exporting to multiple target sites, an object is exported if it was modified since the last export to any site on the list.

Bulk data files are always included and only the latest dataset version is imported.

For assembly synchronization, components that may be owned by sites other than the site from which you are importing an assembly are included. Includes distributed components within a distributed assembly. A distributed assembly consists of components owned by more than one site. First, the top-level assembly and all components owned by the assembly owning site are retrieved. Then individual distributed components are retrieved from their respective owning sites.

Report synchronization state of an object

1. Select the object in My Teamcenter.
2. Choose **Tools→Multi-Site Collaboration→Synchronize→Object**.

Synchronize a component with report only

1. Select the component in My Teamcenter.
2. Right-click and choose **Multi-Site→Synchronization→Object**.
3. Click Synchronization preferences  .
4. Select **Report Only** and, if the component selected is not an item revision, select **Specific Revision Rule** and select a revision rule from the list.
5. Select the other options you desire and click **OK**.

   For information about the options, see the table in *Synchronization on-demand*.
6. In the report pane, select an out-of-date replica you want to update and choose **Tools→Import→Remote** to synchronize it.

Synchronize an assembly with report only

1. Select the assembly in Structure Manager.
2. Right-click and choose **Multi-Site→Synchronization→Assembly**.
3. Click Synchronization preferences  .

4. Select **Report Only**, **Specific Revision Rule**, and select a revision rule from the list.
5. Select the other options you desire and click **OK**.

   For information about the options, see the table in *Synchronization on-demand*.
6. In the report pane, select an out-of-date replica you want to update and choose **Tools→Import→Remote** to synchronize it.

Synchronize a component

1. Select the component in My Teamcenter.
2. Right-click and choose **Multi-Site→Synchronization→Object**.
3. Click Synchronization preferences  .
4. Select **Perform Sync** or **Perform Sync in Background**.
5. If the component selected is not an item revision, select **Specific Revision Rule** and select a revision rule from the list.
6. Select the other options you desire and click **OK**.

   For information about the options, see the table in *Synchronization on-demand*.
7. Perform an on-demand synchronization using the **Report Only** option to verify synchronization of the component occurred properly.

Synchronize an assembly

1. Select the assembly in either My Teamcenter or Structure Manager and right-click and choose **Tools→Multi-Site Collaboration→Synchronization→Assembly**.
2. Select **Perform Sync** or **Perform Sync in Background**.
3. Select the other options you want and click **OK**.

   For information about the options, see the table in *Synchronization on-demand*.

   Note

   From Structure Manager, you can use on-demand synchronization with the **Report Only** option to verify synchronization of the assembly and its components occurred properly.

You can also use the **sync_on_demand** utility to perform these functions from the command line.

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

Automatic synchronization

The user that replicates an object can specify that the replica be synchronized automatically when the master object is modified. The replica will then

be synchronized automatically using Multi-Site Collaboration automatic synchronization features. This results is an efficient and evenly distributed synchronization process and replicas are updated within minutes after the master copy is modified.

Note

Automatic synchronization is not intended to replace the **data_sync** utility. Users can use either the **data_sync** utility, automatic synchronization, or both methods to synchronize data.

To set automatic synchronization options, choose **Options→Edit**.

Option	Purpose
Synchronize Automatically	Synchronizes replica data automatically when the master copy is modified. This option requires the Subscription Manager at both the owning and replica site. This option is not supported when the owning site is running a version earlier than Teamcenter 6.*x*.
Synchronize in Batch Mode	Specifies that replica data is synchronized only when the **data_sync** utility is run at the owning site. This is the default option.
Notify By E-mail	Notifies the user by e-mail when the master object is modified. If notification is requested, a subscription is created at the owning site. In addition, a subscription is also created on the replica; the subscription's handler is a notification handler that ultimately performs the notification at the replica site. This option requires the Subscription Manager at both the owning and replica site.

Note

The first two options in the table, **Synchronize Automatically** and **Synchronize in Batch Mode**, are mutually exclusive. The third option, **Notify By E-Mail**, can be specified in conjunction with the first two options.

If none of these options are selected, the following behavior results:

- If the object is being imported for the first time, the default synchronization is batch mode through **data_sync** with no notification.
- If the object was previously imported, the current option settings are retained, that is, the last options that were selected.

For automatic synchronization to work, the owning site must enable subscriptions by setting the **TC_subscription** site preference to **ON**. For notification to occur, subscriptions must also be enabled at the importing site.

Chapter

25 Support for requirement content

Chapter

25 Support for requirement content

Multi-Site supports performing actions on requirement content in the same manner as other Teamcenter objects. You can import or export requirement objects and its related dataset content (named reference) with ownership transferred to the target site or as a reference (replica) at the target site. You can also synchronize a replicated requirement object and its related dataset content. You can remotely check out, modify, and check in requirement objects. You can use the default **TIEUnconfiguredExportDefault** transfer mode for export and **TIEImportDefault** transfer mode for import of requirements objects.

Appendix

A Glossary

Appendix

A Glossary

A

access control list (ACL)
Access Manager component that contains a list of accessors and the privileges granted, denied, and not set for each accessor.

Access Manager (AM)
Teamcenter application that enables the system administrator to grant users access to Teamcenter objects.

accessor
Access Manager component that grants or denies privileges to clusters of users who share certain common traits (for example, perform the same function or work on the same project).

AM
See *Access Manager (AM)*.

B

BOM
Bill of materials. See also *design bill of materials* and *manufacturing bill of materials*.

BOM view
Teamcenter object used to manage product structure information for an item.

BOM view revision (BVR)
Workspace object that stores the single-level assembly structure of an item revision. Access can be controlled on the structure (BOM view revision) independently of other data. BOM view revisions are meaningful only in the context of the item revisions for which they are created.

bulk data
Physical information represented in the database by a data item. Examples of bulk data are file system items, paper documents, and microfiche. The Teamcenter database describes the bulk data. The bulk data resides elsewhere, for example, in a file system or in a filing cabinet. See also *data item*.

Business Modeler IDE
Teamcenter application that enables a customer to define the following data model objects: business objects, classes, attributes, lists of values, and rules.

D

data item
Teamcenter object representing bulk data defined and manipulated by application products, for example, papers that reside in a filing cabinet, directories and files that reside in a file system, Excel spreadsheets, and CAD model and drawing files. Metadata for the data item resides in the Teamcenter database. See also *bulk data* and *metadata*.

data model
Abstract model that describes how data is represented and used.

dataset
Teamcenter workspace object used to manage data files created by other software applications. Each dataset can manage multiple operating system files, and each dataset references a dataset tool object and a dataset business object.

delivery unit
Subassembly that is manufactured separately and delivered to the assembly plant as a consumed part. One of the operations in the assembly process uses the delivery unit as a consumed part. The components of a delivery unit are not consumed in any of the operations.

design bill of materials
List of components and subassemblies used to define an assembly structure, and the representation of the assembly structure. Compare with *manufacturing bill of materials*.

E

environment variables script
Teamcenter script (**tc_profilevars**) that sets variables for the Teamcenter environment. This script sets all Teamcenter environment variables except **TC_ROOT** and **TC_DATA**.

F

facility
Physical location in an enterprise (for example, manufacturing plant or design center). One facility can comprise multiple sites. Compare with *site*.

FCC
See *FMS client cache (FCC)*.

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's lifecycle visualization. Each Teamcenter rich client host requires a local FMS client cache.

FMS master configuration file
File that configures FMS (**fmsmaster.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 master host
Host that contains the FMS master configuration file (**fmsmaster.xml**). This file is optionally mounted at each FSC server.

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.

folder
Graphical representation of an aggregation of objects, such as a group, class, or subclass. For easy distinction in the class hierarchy, each of these aggregations has a different type of folder icon associated with it: a group folder icon, a class folder icon, or a subclass folder icon.

form
Teamcenter workspace object used to display product information (properties) in a predefined template. Forms are often used to create an electronic facsimile of a hardcopy form in Teamcenter. See also *master form*.

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 timeout value. It can also define default values for FCCs and other FSCs.

G

Globally Unique Identifier
Identifier that is assigned to each I-deas data item by the I-deas applications software and used to identify the data item. See also *Item GUID* and *version GUID*.

GUID
See *Globally Unique Identifier*.

I

IDSM server
Integrated Distributed Services Manager, a network node that runs a daemon process to handle the transfer of data objects among sites in a Multi-Site Collaboration network. One IDSM server node must be designated for each Teamcenter database from which objects are published; each server node can act for one database only.

instance
Single data object that is associated to a class. The instance can correspond to a line in the BOM.

item
Workspace object generally used to represent a product, part, or component. Items can contain other workspace objects including other items and object folders.

Item GUID
Common GUID assigned to all versions of an I-deas data item in the data item series. See also *GUID* and *version GUID*.

item relation
Description of an association between a Teamcenter item and a piece of information that describes or is related to the item.

item revision
Workspace object generally used to manage revisions to items.

item revision relation
Description of an association between a Teamcenter item revision and a piece of information that describes or is related to the item revision.

K

key
Set of attributes defining a unique user level identifier for a class.

M

manufacturing bill of materials
Defines how the product is manufactured, rather than how it is designed. Compare with *design bill of materials*.

master form
Teamcenter workspace object used to display product information (properties) in a predefined template. Master forms are used to display product information in a standardized format.

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.

master object
The controlling object in a Multi-Site Collaboration network. When an object is created and saved, that instance is the master object until it is exported with transfer of ownership. There can be only one master object in a Multi-Site Collaboration network, and only the master object can be modified. If a master object is replicated, it cannot be deleted until all replicated objects are deleted.

metadata
Object description in the Teamcenter database.

Multi-Site Collaboration
Teamcenter capability that allows the exchange of data objects among several Teamcenter databases. Transfer of objects among databases is controlled by daemon processes running on designated servers. Objects are replicated by exporting them from their original database and importing them into the requesting database. Configuration of Multi-Site Collaboration is optional.

Multi-Site Collaboration network
Network of independent Teamcenter sites that are within the same enterprise and share data using Multi-Site Collaboration.

My Teamcenter
Teamcenter rich client application that is the main access point for managing product information. My Teamcenter provides the functionality for creating objects in the Teamcenter database, querying the database for objects, checking in and checking out objects, and managing tasks. Users can also open objects, automatically launching the related application.

Each user has a personal My Teamcenter window that displays product information as graphical objects. Although users share product information across the enterprise, they organize this information individually in personal workspaces.

N

named ACL
Named group of access controls. See also *access control list (ACL)*.

named reference
File types that are managed by a dataset. Datasets are the only workspace objects that use named references.

O

object directory services server
Multi-Site Collaboration network node that runs a daemon process to handle publication of data objects within a Multi-Site Collaboration environment. One ODS server node must be designated for each object directory services site and each server node can act only for one object directory services site.

object directory services site
Site with the database that maintains a record of each object in a Multi-Site Collaboration network. At least one Teamcenter database on a Multi-Site Collaboration network must be designated as an ODS site. This site is used to store publication records for the data objects.

ODS server
See *object directory services server*.

ODS site
See *object directory services site*.

Oracle server
Single installation of Oracle able to service queries from several Teamcenter workstations. The **ORACLE_SERVER** environment variable defines this Oracle service node. For large-scale installations, the Oracle server is typically a dedicated high performance workstation that is optimized specifically for running Oracle software.

Organization
Teamcenter application that enables a system administrator to create and manage critical Teamcenter files and database entries. It is the point of access for creating a company's virtual organization and for performing system administration activities such as volume creation, maintenance, and site administration. Organization enables creation and management of person, user, role, and group definitions; definition of the hierarchical structure of the Teamcenter organization; management of data volumes; and establishment and maintenance of Teamcenter sites.

Organization List Tree
List format display of the components of the organization: groups, roles, users, and persons. Creation and maintenance of these components, as well as sites and volumes, can be performed by selecting a node from the list and performing functions in the corresponding dialog box.

organization tree
Graphic display of the Teamcenter organization structure. Expanding and collapsing branches of the tree enables viewing and managing the organizational structure. Selecting a node starts Organization wizards used to create groups, subgroups, roles, users, and persons.

owner
User that owns an object, initially the user who created it. Ownership can be transferred from the owner to another user. An object owner usually has privileges that are not granted to other users (for example, the privilege to delete the object).

owning group
Group that owns an object, usually the group of the user creating the object. Because users commonly share data with other members of a group, additional privileges may be granted to the owning group (for example, the privilege to write to the object).

owning site
Multi-Site Collaboration site where the master object resides. The owning site is the only site where the object can be modified.

P

persistent object manager (POM)
Interface between Teamcenter objects and the Relational Database Management System (RDBMS). The persistent object manager provides definition of classes by inheritance from other classes and definition of attributes, manipulation of in-memory objects and support for their saving and retrieval to and from the underlying RDBMS, support for applications accessing the same data concurrently, protection against the deletion of data used by more than one application, and support for the access control lists attributed to objects.

PLM XML
Siemens PLM Software format for facilitating product life cycle interoperability using XML. PLM XML is open and based on standard W3C XML schemas. Representing a variety of product data both explicitly and via references, PLM XML provides a lightweight, extensible, and flexible mechanism for transporting high-content product data over the Internet.

POM
See *persistent object manager (POM)*.

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.

preference scope
Hierarchical range for which a Teamcenter preference can be set. The scope of a preference can be site, group, role, or user.

process
Automation of a business procedure, describing the individual tasks and task sequences required to complete a business procedure.

Product Data
Teamcenter application used to access and interact with the items and item revisions that represent a company's products, parts, and components.

product item
Top-level item associated with an assembly structure representing a generic product, for example, a midsize sports utility vehicle program for a given model year.

product structure
Hierarchy of assembly parts and component parts with a geometric relationship between them, for example, a bill of materials (BOM). Variant and revision rules define the generic BOM. This BOM can then be loaded to display the configured variant.

product view
Saved configuration of the assembly viewer, including the selection of objects, zoom factor, rotation angle, and pan displacements.

project
Basis for identifying a group of objects available to multiple organizations, such as project teams, development teams, suppliers, and customers for a particular piece of work.

published object
Object available to other sites in a Multi-Site Collaboration network. Publishing an object creates a record in the ODS site database that can be read and searched by other sites. Until an object is published, it can be seen only by the owning site.

R

relation
Description of an association between a Teamcenter object and a piece of information that describes or is related to the object.

release status
Status associated with a workspace object when it is released through a workflow process.

replicated object
Copy of master object residing at sites within a Multi-Site Collaboration network. See also *master object*.

revision rule
Parameter set by the user that determines which revision of an item is used to configure product context.

rich client
Java-based user interface to Teamcenter installed on user workstations. The rich client accesses Teamcenter databases using a remote or local server.

role
Function-oriented cluster of users that models skills and/or responsibilities. The same roles are typically found in many groups. In Access Manager, role is an accessor used to grant privileges to all users with the same skills and/or responsibilities regardless of project.

S

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 preference
Teamcenter preference that applies to the entire site.

subassembly
Assembly that is built into the assembly structure of another assembly or intended for that use. In a manufacturing view, either a delivery unit or a workpiece. See also *delivery unit* and *workpiece*.

subscription
Combination of a workspace object and event to which a Teamcenter user requests notification of occurrence. Teamcenter notifies a subscribed user when the event occurs in association with the object. Users can subscribe to objects from Teamcenter applications, such as My Teamcenter and Structure Manager.

Subscription Manager
Tool used to find, delete, and modify active subscriptions.

Subscription Monitor
Teamcenter application that enables a system administrator to query subscriptions created in the database, to monitor and delete subscription events and actions, and to generate statistical reports in either text or bar chart format regarding subscriptions, events, and actions.

supersedure
Manually created relation that graphically displays deleted components and the components that replace them. A supersedure is always created in the context of a parent assembly. Thus a single component can be used in more than one supersedure if it is used in different parent assemblies. A supersedure can be created for changes of part number or of quantity, but not for changes in a part revision.

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.

top level
Object at the root of a product structure where a process plan is being developed. The top level can be either an end product being manufactured or a subassembly used in the end product (for example, an engine for a tractor where the tractor is the end product).

U

unpublished object
Object not available to other sites in a Multi-Site Collaboration network. Users can unpublish previously published objects so they are once again accessible only to the owning site.

user preference
Teamcenter preference applying to a specific user.

V

version GUID
Distinct GUID assigned to each separate version of an I-deas data item in the data item series. The version GUID uniquely identifies the data item.

view
Tailored representation of objects within a class. Views are associated with abstract and storage classes. Attribute properties can also be applied. For example, a class might define the physical and accounting attributes for its objects, but a view for tool designers might display only the physical attributes, and a view for accountants might display only pricing and order number attributes.

visualization
Ability to display a realistic, real time, graphical visualization of geometric data.

W

Web Browser
Teamcenter application that provides access to Internet Web pages from within the rich client framework. The Web browser is a rich client window that acts as a Web browser, enabling you to navigate and view Web pages within the rich client rather than switching to a separate Web browser. The Web browser also provides the ability to access MIME (Multipurpose Internet Mail Extension) file types and to view files created in other applications, such as Microsoft Word and Excel, through the Web browser.

working revision
Revision that can be changed by a user with write privileges. No record of intermediate states of a working revision is maintained by Teamcenter.

workpiece
Intermediate state of the product during the manufacturing process. In each step of the manufacturing process, the workpiece is positioned in the work area and the work instructions are performed. The resulting workpiece then flows to the next operation in the sequence, where the next operation is performed.

Index

D

P

R

S

T

U

V

W

X