TEAMCENTER

Multi-Site Collaboration

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

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

Support Center: **support.sw.siemens.com**

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

Contents

Part II: Configuring and administering Multi-Site Collaboration

Planning and setup

System administration

Custom configurations

Setup verification

Managing Multi-Site Collaboration behavior

Part III: Using Multi-Site Collaboration

Publishing and unpublishing

Updating an object or BOM

Using synchronization

Archiving and restoring data

Using a remote inbox

Prepopulate a target FSC for global data cache

Troubleshooting Multi-Site

Part I: Introduction

Use Multi-Site Collaboration to easily share product information across your enterprise. Multi-Site Collaboration allows the exchange of Teamcenter data objects between databases, providing semi automated real-time data sharing across the entire enterprise.

Multi-Site Collaboration is one of several solutions Teamcenter offers to meet different data sharing needs.

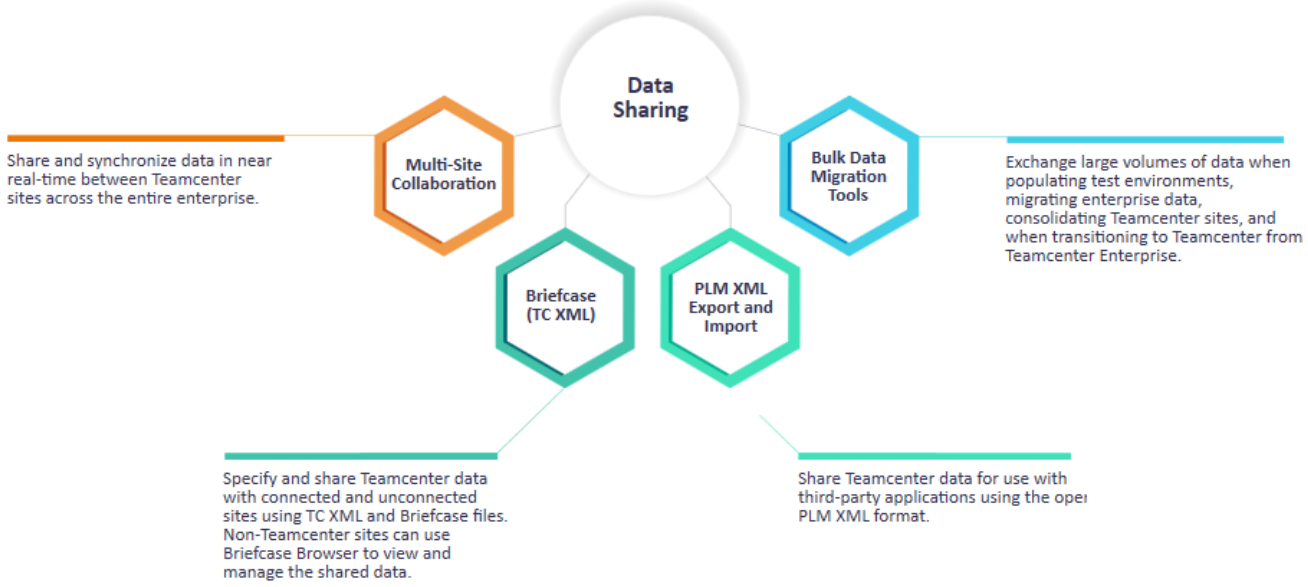


Table 1-1. Where do I go from here?

Business User	
Learn more about data replication.	See **Data replication**.
Share objects with other sites.	See **Viewing objects that are visible to other sites**.
Import remote objects.	See **Finding objects to import**.
Modify remote objects	See **Obtaining write access to shared objects**.
Administrator	
Set up Multi-Site Collaboration.	See **Planning and setup process**.
Learn more about administering Multi-Site Collaboration.	See **Requirements for Multi-Site system administrators**.
Resolve export and import issues.	See **Debug remote import/export problems**.
Migrate from legacy Multi-Site Collaboration to using TC XML-based Multi-Site Collaboration.	See **TC XML-based Multi-Site Collaboration transition overview**.

1. Getting started with Multi-Site Collaboration

Basic Multi-Site Collaboration requirements

To allow Multi-Site Collaboration to exchange Teamcenter data objects between databases, each database must be easily accessible using TCP/IP, either over the Internet or over your company intranet. Be aware of the following items:

- 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 each site's preference file.
- Identify the network nodes you want to run Multi-Site Collaboration server processes for these databases and configure those systems to run the processes.

Requirements for using Multi-Site Collaboration

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. See **Enabling Multi-Site Collaboration**

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. See **Configuring and administering Multi-Site Collaboration**.

2. Enabling Multi-Site Collaboration

Install and enable Multi-Site Collaboration

Multi-Site Collaboration allows the exchange of Teamcenter data objects between databases. Each database should be easily accessible via TCP/IP, either over the Internet or the company intranet. Configuration of Multi-Site Collaboration is optional.

Coordinate configuration of Multi-Site Collaboration with the system administrators of the other Teamcenter databases to be part of the 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, identify the network nodes to run Multi-Site Collaboration server processes for these databases and configure those systems to run the processes.

Be aware that if an environment is designated a test environment, it cannot be configured for Multi-Site Collaboration. The environment type must be changed before configuring it for Multi-Site Collaboration.

Prepare the Multi-Site Collaboration environment

Perform the following steps to configure Multi-Site Collaboration for a wide area network:

1. Identify all Teamcenter databases to be part of the Multi-Site Collaboration environment.

2. Identify the Teamcenter database to act as the ODS database.

   This database stores records about the data objects published by other databases in the Multi-Site Collaboration environment (that is, made public to the other databases).

   This can be one of the databases identified in step 1 or it can be a dedicated database. The database must be populated with Teamcenter data.

3. For each database identified in step 2, identify a network node local to that database to act as the ODS server.

   The **ods** service (or daemon) runs on this system to listen for publication queries from other databases.

4. For each database identified at step 1, identify a network node local to that database to act as the IDSM for that database.

   When other databases request an object published from this database, the **idsm** service (or daemon) is run on this network node to export the object.

5. For each database identified in step 1, obtain the site name and site ID.

The site ID of the database is generated during installation and cannot be changed. The site name is customizable but by default is based on the site ID. To obtain the site name and site ID, use the administration application named **Organization** in Teamcenter rich client (in the rich client application manager, click **Admin** and then click the **Organization** symbol). Within **Organization**, choose the top-level **Sites** node from the **Organization** tree. The site details for the local database are listed first.

6. Using the information obtained in steps 2 through 5, populate each database site table with information about the other sites using the Organization application in the Teamcenter rich client.

   The node for each site is the name of the network node to run the necessary Multi-Site Collaboration services (or daemons) (**idsm** and/or **ods**). If the site is an ODS database, check the ODS site flag. To publish objects from the ODS database, define the site of the ODS database in the site table and configure the ODS server as an IDSM server.

7. For each database identified in step 1 and step 2, edit the site preference for the database and modify the following preferences to reflect the Multi-Site Collaboration environment:

   ODS_permitted_sites (ODS database only)
   ODS_site (Non-ODS databases)
   ODS_searchable_sites
   ODS_searchable_sites_excluded
   IDSM_permitted_sites
   IDSM_permitted_users_from_site_*site-name*
   IDSM_permitted_transfer_sites
   IDSM_permitted_transfer_users_from_site_*site-name*
   IDSM_permitted_checkout_sites
   IDSM_permitted_checkout_users_from_site_*site-name*
   Fms_BootStrap_Urls
   TC_publishable_classes
   TC_transfer_area

8. For each database identified in step 1 and step 2, copy all POM transmit schema files for that database into the POM transmit schema directories for each of the other databases.

   This step is required to allow the import of data objects from other databases. Devise a strategy for regularly synchronizing POM transmit schema directories.

9. For each network node identified at step 3 and step 4, run the Teamcenter setup (Windows) or installation (Linux) program on that node to configure and start the Multi-Site Collaboration services or daemons.

Install a proxy server

Configure a proxy server to be used with Multi-Site Collaboration.

Configure Multi-Site Collaboration daemons (Linux)

Configure the Multi-Site Collaboration daemons:

1. As a user with root privileges, run the **root_post_tasks_*id*.ksh** program in the **install** directory in the Teamcenter application root directory.

2. At the command line, execute the following command:

   ```
   ps -ef | grep -v grep | grep xinetd
   ```

   This script obtains the current process ID of the **xinetd** daemon.

3. At the command line, execute the following command:

   ```
   kill -HUP process-id
   ```

   Replace *process-id* with the **xinetd** daemon ID obtained in step 2.

This procedure adds the **idsm** daemon entry to the **xinetd.conf** file and forces the **xinetd** daemon to reload its configuration. As a result, the Multi-Site Collaboration daemons are launched to complete the installation.

Optionally migrate Teamcenter data using Multi-Site

The Multi-Site **data_share** utility supports bulk migration of Teamcenter data between different Teamcenter releases when, for example, moving from an earlier on-premises Teamcenter release to a later cloud-based release. This process requires that the source and target sites must be running Teamcenter 11.6 or later with TC XML-based Multi-Site installed. See Migrating Teamcenter data to a newer version of Teamcenter for details.

3. Upgrading from a previous version

Considerations when upgrading from legacy Multi-Site Collaboration

If you are upgrading a site that has legacy Multi-Site Collaboration installed, be aware that legacy Multi-Site Collaboration is no longer supported. When upgrading to this release of Teamcenter, TC XML Multi-Site Collaboration is automatically enabled.

Refer to **TC XML-based Multi-Site Collaboration transition overview** for details on upgrading.

Prepare a Multi-Site Collaboration site for upgrade

Check for and recover transaction failures resulting in unrecovered transfer locks two or three days prior to the upgrading the site. All services can continue to run during this process.

1. Run the **ensure_site_consistency** utility with the **report** function at the upgrade site:

   ```
   ensure_site_consistency -u=tc-admin-user -p=group -g=group
      -f=report -search -report=myReport.txt
   ```

   Check the output file for possible transfer failures.

   ```
   ** Command line option used to generate the Report **
   -f=report -search -report=myReport.txt

   Objects Found: 0

   ------------------------------------------------------------------------------
   Object Id / Name   Class Name of Object     Owning Site     Action/Failure
   ------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   Object             Class Name               Owning Site     Action/Failure
   ------------------------------------------------------------------------------
   TEST05             Item                     tc101d1         At this site, run
                                                               ensure_site_consistency
                                                               -f=recovery -search
   TEST04             Item                     tc101d1         At this site, run
                                                               ensure_site_consistency
                                                               -f=recovery -search
   ```

2. If you find possible failures, use an SQL command to query the creation date of the SST recovery datasets (**TC_sst_record**), for example:

   ```
   select pcreation_date, rsecondary_objectu from PIMANRELATION,
   PPOM_APPLICATION_OBJECT pao where rsecondary_objectu = pao.puid and
   rrelation_typeu = (select puid from PIMANTYPE where ptype_name =
   'TC_sst_record') order by pcreation_date;
   ```

The output of this query shows the creation data for each SST dataset and the dataset unique identifier (UID), for example:

```
PCREATION_DATE,RSECONDARY_OBJECTU
2013-02-20 01:39:13.0,qYaFy0UbY54p9B
2013-02-28 01:45:17.0,qYdFiby4Y54p9B
2013-02-28 01:47:46.0,qYaFibSbY54p9B
```

> Note:
>
> The time period for determining an active or aborted transaction is based on the assumption that only parts or moderate-size assemblies are selected for export with transfer of ownership. Exporting a very large assembly with ownership transfer, such as an entire vehicle, is not an expected or recommended practice.

- If the SST dataset creation date is within the last 24 hours, you may ignore the transaction as it may be active and should complete prior to the start of the scheduled upgrade.

- If the SST dataset creation date is more that 24-hours-old, the transaction has aborted. To avoid the delay caused by open transfer locks during the upgrade operation, run the **ensure_site_consistency** utility with the **recovery** function:

  ```
  ensure_site_consistency -u=tc-admin-user -p=password -g=group
     -f=recovery -search -report=myReport.txt
  ```

  Optionally, you can run this command after the upgrade operation completes. However, this may cause delays during the upgrade and requires an administrator to monitor the process for required actions.

  > Caution:
  >
  > Siemens Digital Industries Software highly recommends that you do *not* defer the recovery of transfer locks. During the *Prevent site consistency failures during upgrade* procedure, you are instructed to remove all transfer locks. There is a possibility of objects changing during the time that the transfer locks are missing. Running the recovery function at this time minimizes the time period the objects are exposed to the risk of the objects being locked by a process other than the upgrade process.

Prevent site consistency failures during upgrade

Twenty-four hours before performing these steps, inform your users that Multi-Site transfers are prohibited pending the completion of the upgrade process. Then, shut down the Integrated Distributed Services Manager (IDSM) and Object Directory Services (ODS) for the site. Doing so minimizes the possibility of open transfer locks being encountered during the upgrade or that you may have to wait for an active transaction to complete before performing the upgrade operation.

1. Run the **ensure_site_consistency** utility with the **report** function at the upgrade site:

```
ensure_site_consistency -u=tc-admin-user -p=password -g=group
   -f=report -search -report=myReport.txt
```

Check the output file for possible failures.

```
** Command line option used to generate the Report **
-f=report -search -report=myReport.txt

Objects Found: 0

-------------------------------------------------------------------------------
Object Id / Name   Class Name of Object     Owning Site     Action/Failure
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Object             Class Name               Owning Site     Action/Failure
-------------------------------------------------------------------------------
TEST05             Item                     tc101d1         At this site, run
                                                            ensure_site_consistency
                                                            -f=recovery -search
TEST04             Item                     tc101d1         At this site, run
                                                            ensure_site_consistency
                                                            -f=recovery -search
```

2. If you find possible failures, use an SQL command to query the creation date of the SST recovery datasets (**TC_sst_record**), for example:

```
select pcreation_date, rsecondary_objectu from PIMANRELATION,
PPOM_APPLICATION_OBJECT pao where rsecondary_objectu = pao.puid and
rrelation_typeu = (select puid from PIMANTYPE where ptype_name =
'TC_sst_record') order by pcreation_date;
```

The output of this query shows the creation data for each SST dataset and the dataset unique identifier (UID), for example:

```
PCREATION_DATE,RSECONDARY_OBJECTU
2013-02-20 01:39:13.0,qYaFy0UbY54p9B
2013-02-28 01:45:17.0,qYdFiby4Y54p9B
2013-02-28 01:47:46.0,qYaFibSbY54p9B
```

> Note:
>
> The time period for determining an active or aborted transaction is based on the assumption that only parts or moderate size assemblies are selected for export with transfer of ownership. Exporting a very large assembly with ownership transfer, such as an entire vehicle, is not an expected or recommended practice.

- If the SST dataset creation date is within the last 24 hours, wait for the transaction to complete.
- If the SST dataset creation date is more that 24-hours-old, the transaction has aborted. The aborted transactions can be recover after the upgrade completes using the **ensure_site_consistency** utility.

3. Change the Oracle listener port for the site database to avoid accidental connections during the upgrade process.

4. Run SQL commands to remove all transfer locks, for example:

```
delete from PM_PROCESS_LOCK;
update POM_M_LOCK set column-name-expression where lock_mode is null;
```

5. (Optional) Run the Teamcenter **clearlocks** utility to clear any process lock, for example:

```
-clearlocks -u=tc-admin-user -p=password -g=group -assert_all_dead
```

Upgrading a Multi-Site Collaboration site

Before upgrading the site using Teamcenter Environment Manager (TEM), complete the tasks listed in *Prepare a Multi-Site Collaboration site for upgrade* and *Prevent site consistency failures during upgrade*. Doing so ensures that the site consistency check does not cause any issues with the upgrade.

Complete a Multi-Site Collaboration site upgrade

After **upgrading Multi-Site Collaboration site**, if you deleted any transfer locks during the final step of *Prevent site consistency failures during upgrade* tasks:

1. Run the **ensure_site_consistency** utility with the **restore_transfer_locks** function to restore the transfer locks, for example:

```
ensure_site_consistency -u=tc-admin-user -p=password -g=group
-f=restore_transfer_locks -search –report=myReport.txt
```

2. Run the **ensure_site_consistency** utility with the **recovery** function to attempt to recover the failed transactions.

Generate upgraded site dataset mapping files

When you upgrade a site that participates in Multi-Site Collaboration, you must generate a new dataset mapping file for each site defined at the upgraded site and a new dataset mapping file for the upgraded site at each participating site. If the participating site is Teamcenter 8 or later, you can use the **database_verify** utility to generate the dataset mapping file.

1. Run the **database_verify** utility at the upgraded site with the **-site** argument value set to **ALL**, for example:

```
database_verify -u=tc-admin-user -p=password -g=group -site-ALL
```

2. Run the **database_verify** utility at each Teamcenter 8 or later site with the **-site** argument value set to the upgraded site, for example:

`database_verify -u=`*tc-admin-user* `-p=`*password* `-g=`*group* `-site-`*upgraded-remote-site*

TC XML data exchange compatibility for Product Configurator

Teamcenter Product Configurator TC XML data cannot be exchanged across all Teamcenter versions. A general limitation of TC XML data exchange is that, whenever there are major data model changes across Teamcenter versions, TC XML data cannot be exchanged across those versions. Because Teamcenter had major data model changes in version 11.6, any TC XML data exported from prior versions cannot be imported to Teamcenter 11.6 or later.

Follow these guidelines when upgrading Multi-Site:

- While performing the upgrade at the replica site, do not upgrade the variant rule object to the new expression data model. This is to ensure that site ownership is not changed for the variant rule object.
- Run the **data_sync** utility before using the variant rule object at the replica site. Doing so upgrades the variant rule to the new expression data model.
- To avoid conflicts, use the site specific naming rule for all configurator objects while running the **data_sync** utility.
- Before the upgrade process, update or delete the variant criteria object with the duplicate thread ID. The MFK of the variant criteria is modified to include the thread ID. The Teamcenter upgrade fails if variant criteria objects with the duplicate thread ID exist in the system. You must update or delete the variant criteria object with the duplicate thread ID before the upgrade process.

Version interoperability

Multi-Site Collaboration is interoperable with sites running a limited set of earlier releases. Therefore, when you upgrade Multi-Site Collaboration site to a new version, you do not need to upgrade all other sites in the Multi-Site Collaboration network at the same time. Refer to the **Support White Papers Certifications** page for releases supported by your version of Multi-Site Collaboration.

Version compatibility

Although version 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 sites can be multilingual sites. Multilingual 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.

Restrictions

Be aware of the following restrictions:

- While Siemens Digital Industries Software supports Multi-Site Collaboration interoperability between the specified earlier releases, sites running a previous release of Multi-Site Collaboration cannot import classes introduced in later releases. Attributes that do not exist on a site running an older release of Multi-Site Collaboration will not be imported. By default, extra attributes that exist on an importing site will be set with the default value.

  With **TC XML-based Multi-Site Collaboration** enabled, you can configure closure rules to bypass those classes. This approach allows you to avoid import failures at sites not supporting certain classes. You can also use the Advanced Multi-Schema Exchanger to convert the XML content before importing it.

- New attributes in the current version of Teamcenter added to POM classes that existed in earlier versions are exported from Teamcenter, but are not imported into earlier versions. When you import these attributes from earlier versions to the current version of Teamcenter, they are assigned null values.

- If a new type is added in the current version of Teamcenter, for example, 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 such as Business Modeler IDE. You can also choose to use closure rules to avoid exporting the types.

- 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 features to be temporarily disabled until all sites are upgraded to the current version of Teamcenter.

  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.

General Requirements

Be aware of the following general requirements applicable in Teamcenter:

- 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 not be sent at all. You can configure this behavior using closure rules.

  For example, to include the relation during remote check-in:

  ```
  CLASS.ItemRevision:CLASS.Dataset:RELATIONP2S.RelationType_A:PROCESS+TRA
  VERSE:$opt_remote_checkin==”true”
  ```

  To exclude the relation during remote check-in:

  ```
  CLASS.ItemRevision:CLASS.Dataset:RELATIONP2S.RelationType_A:SKIP:$opt_r
  emote_checkin==”true”
  ```

  For additional information about working with closure rules, see PLM XML/TC XML Export Import Administration.

4. Migrating Teamcenter data using Multi-Site

Migrating Teamcenter data to a newer version of Teamcenter

The Multi-Site **data_share** utility supports bulk migration of Teamcenter data between different Teamcenter releases when, for example, moving from an earlier on-premises Teamcenter release to a later cloud-based release. You have the following tools available to assist you in this process.

- You can run the **tc_customization_compare_report** before the migration to identify BMIDE customizations. Evaluate and address how these customizations will be handled at the target site.
- You can **run the t2c_schema_analyzer_report utility** before the migration to analyze the data model templates of the two installations to identify any changes needed in the source site Teamcenter data to be compliant with the target site's data model. Addressing those changes enables you to avoid the need to upgrade the source site to the same version as the target site before the migration.

Prerequisites

When the source and target Teamcenter sites are not the same version, the following process requires that the migration utilities on the earlier (source) site be updated to the same level as the utilities on the later (target) site. Update the source site utilities using the process covered in **Deliver the migration utilities to the source (earlier) Teamcenter site**.

Contact your Siemens Digital Industries Software representative if you have any questions about bulk migration of Teamcenter data.

Migrating Teamcenter data

The process of migrating Teamcenter data to a newer version of Teamcenter has the following phases:

1. **Identify and accommodate BMIDE customizations**.
2. **Analyze the data models on the two sites**.
3. **Address customizations and data model differences**.
4. **Migrate the data from the source site to the target site**.
5. **Review the data migration**.

Deliver the migration utilities to the source (earlier) Teamcenter site

When the source and target Teamcenter sites are not the same version, migrating Teamcenter data to a newer version of Teamcenter requires that the migration utilities on the earlier (source) site be updated to the same level as the utilities on the later (target) site. Use the **t2c_create_installer** utility and the following processes to update the source site utilities.

Contact your Siemens Digital Industries Software representative if you have any questions about bulk migration of Teamcenter data.

Package the utilities on the target (later release) Teamcenter site

Run the following command from a Teamcenter command window on the target (later release) site to package the files needed to update the utilities at the source (earlier) site.

```
t2c_create_installer -f=create_installer -installer_location=output_dir
```

Where *output_dir* is the folder in which the files needed to deliver to the target site are saved. The files are saved in a file named *InstalledData.zip*.

Deliver the *InstalledData.zip* file to the source site.

Update the utilities on the source (earlier release) Teamcenter site

The following steps assume you are running the analyzer utility on the target site. See t2c_schema_analyzer_report for the arguments to use when running the utility on the source site.

1. Unzip *InstalledData.zip* to a directory on the source site to create, for example, *c:\temp\InstalledData*.

2. Copy the following files to *TC_Root\bin*.

   t2c_create_installer.bat (for windows)

   t2c_create_installer.sh (for Linux)

   t2c_create_installer.perl

3. Open a Teamcenter command window and run the following command.

   ```
   t2c_create_installer -f=deploy -installer_location=input_dir
   ```

   Where *input_dir* is the folder containing the *InstalledData* folder, for example, *c:\temp*.

   The source site is updated to run the current migration utilities.

Identify and address BMIDE customizations

Use the following steps to analyze and report the BMIDE differences between two Teamcenter sites. Reviewing these reports and addressing any differences is particularly useful when moving from an earlier on-premises Teamcenter release to a later cloud-based release.

With this process, you use the **tc_customization_compare_report** utility to generate packages of data model and template files for each site. You then use the utility to compare those packages and generate reports of any differences between the sites.

1. In Teamcenter command windows on the source (earlier on premises) and target (later cloud-based) sites, run the following command.

   ```
   tc_customization_compare_report -f=create_package -type=bmide -out=dir
   ```

   Where *dir* is the folder in which the generated package (folder) of model and template files is stored.

2. Copy the source site package directory to the target site.

3. In a Teamcenter command window on the target site, run the following command. You must be a Teamcenter administrator with DBA privileges to run this command.

   ```
   tc_customization_compare_report -u=admin-username -p=password
   -f=generate_report -type=bmide -target=target_dir
   -source=c:\temp\source_dir -report_folder=report_dir
   -sourceVersion=source_ver -targetVersion=target_ver
   ```

   Where:

   admin-username **and** *admin-username*

   Credentials of a user with Teamcenter administrator and DBA privileges.

   target_dir

   The full path of the directory containing the target (later) package directory.

   source_dir

   The full path of the directory containing the source (earlier) package directory.

   report_dir

   The folder in which the generated report files are saved.

   source_ver

   The version of Teamcenter on the source site. Use the form **TC***release*. For example, **TC2312** or **TC2406**.

   target_ver

The version of Teamcenter on the target site. Use the same form as described for *source_ver*.

The reports are generated and saved in *report_dir*.

4. Open *report_dir\t2c_applicationReport.html* to view the generated reports. Evaluate these differences and determine how to address them at the target site.

Analyze the source data for compliance with the target data model

Use the following steps to analyze the data model templates of the source and target sites and identify any changes needed for the source site data to be compliant with the target site's data model.

Requirements

- The **t2c_schema_analyzer_report** requires access to the source and target sites' data model directories (or copies of their contents.)
- The source and target sites must be running Teamcenter 11.6 or later with TC XML-based Multi-Site installed.

Obtain copies of one or both site data model templates

The data analysis utility requires access to the data model templates of the source and target sites.

Copy the contents of the following directories from both of the sites and save them locally. The template files are in the roots of the source and target sites' *tc-data*`\model` directories. (Alternately, if you have operating system access to the sites, you can analyze the files in the installation directories directly using the **t2c_schema_analyzer_report** utility's **-source** and **-target** arguments.)

Profile the data on the source site

You can create a profile of the business objects used at the source site. This information is used by the **t2c_schema_analyzer_report** utility to more precisely identify changes needed to the source site data. You can opt to not profile the source site data, but doing so will result in the need for manual analysis of significantly more object differences between the databases. Generate the profile as follows:

1. At the source site, run the following command:

   ```
   t2c_schema_analyzer_report -db_profiler_sql
   ```

   A sample SQL query is output to the command window.

2. Run the output SQL query.

   The script creates a *.csv* file named for the source site's version of Teamcenter and the current date.

Analyze the data

The following steps assume you are running the analyzer utility on the target site. See t2c_schema_analyzer_report for the arguments to use when running the utility on the source site.

1. Open a Teamcenter command window on the target site.

2. Enter a command using the following form:

   ```
   t2c_schema_analyzer_report.bat -version=from_version
   -report_folder=folder_name -source=source_templates
   -db_profiling_file=profile_file
   ```

   Where:

   from_version

   The Teamcenter version at the source site. Use the form **TC***major***.***minor*. For example, **TC12.3** and **TC 13.1**. You can further include patch version numbers if the site has been patched. For example, **TC12.3.0.1**. Refer to the Help > About box for the version.

   folder_name

   The full path of the directory in which the report and batch files are saved.

   source_templates

   Specifies the directory containing the source site's template files.

   profile_file

   Specifies the data profile *.csv* file created in *Create a profile of the source site's data* earlier in this topic.

   When the utility completes its analysis, it reports the run results in the command window and saves several files in the directory specified by **-report_folder**.

   - *t2c_schema_analyzer_Report.xml* summarizes the business object and attribute changes need for the source site data to be compatible with the target site's data model.

   - *t2c_schema_analyzer_Report.csv* provides a detailed breakdown of the business object and attribute changes needed for the source site's data to be compatible with the target site's data model.

     See **Update the source data as necessary** for information on working with the schema analyzer reports and recommendations on addressing source site data issues.

   - *t2c_post_migration* is a batch file that finalizes the migration. Run this script after **migrating your data**.

Update the source data as necessary

The files created using the **t2c_schema_analyzer_report** utility include the *t2c_schema_analyzer_Report.csv* file. This file provides a detailed breakdown of the business object and attribute changes needed for the source site's data to be compatible with the target site's data model once migrated to the target site. The related *t2c_schema_analyzer_Report.xml* summarizes the changes needed.

Open *t2c_schema_analyzer_Report.csv* in Microsoft Excel or other application. The information in the report is organized by template and business object. For each business object, each of its attributes is also analyzed.

For each business object, the object and attributes, the report describes the differences between the objects in the source and target data models, and whether an action is required for the object to be compatible with the later release. If an action is required, the type of action is stated.

Following are typical actions and recommendations for performing them.

Custom action needs to be identified

Review the report details for the entry. Following are typical scenarios.

Data difference	Recommended action
Class or attribute has been removed.	Use Advanced Multi-Schema Exchanger to map the missing class or attribute to one that exists at the target site. There is no need to map the class or attribute if they are not required at the target site.
Attribute on source site null allows a null value, but the target site does not allow a null value.	Run a SQL query on the source database to select the attribute value. Ensure there is not a null value for the attribute.
Attribute on source site has **isUnique** set to false, but the target has **isUnique** set to true.	Run a SQL query on the source database to select the attribute value. Ensure all values of that attribute are already unique.

Execute t2c_post_migration script

The change is handled by running the **t2c_post_migration** script on the target site after you have migrated your data from the source site to the target site. See **Migrate the source site data to the target site** for more information.

Migrate the source site data to the target site

After **reviewing and addressing issues** identified by the **t2c_schema_analyzer_report** utility, use the following processes to migrate the source data to the target site.

Register the data you will migrate from the source site to the target site

Before migrating data to the target site, register the data to be migrated using the Multi-Site **data_share** utility. Doing so will allow for monitoring the data migration status as described in **Review the data migration**

1. Open a Teamcenter command window on the source site.

2. Register the objects you plan to migrate using the **-function=register_migration_data** option as in the following example.

   ```
   data_share -function=register_migration_data
   -filename=d:\export\item.txt
   ```

   This example presumes the items to migrate are listed in the *item.txt* file. See **data_share** for other options to specify objects to migrate.

Migrate data from the source site to the target site

Use the **data_share** utility with the **-optionset** argument to migrate the source site data to the target site.

- If you are running the utility from the source site, use the **-f=send** function or **-f=offline_export** function as appropriate.

- If you are running the utility from the target site, use the **-f=remote_import** function or **-f=offline_import** function as appropriate.

Review the results of the migration as described in **Review the data migration**. Correct any issues and rerun the migration.

Run the post-migration script

After migrating the source data, run the **t2c_post_migration** script on the target site to finalize the migration.

1. Open a Teamcenter command window on the target machine after the data has been migrated with **data_share**.

2. Run the **t2c_post_migration** script using the following form:

   ```
   t2c_post_migration -u=Tc-admin-user -p=password -g=group
   ```

   The data migration is complete.

Review the data migration

Use the **t2c_report_extract** and **t2c_report_comparator** utilities as follows to generate migration status and object differences reports. Review the reports, address any migration issues, and re-run the migration as necessary.

Prerequisites

Register and then migrate the data to the target site as described in **Migrate the source site data to the target site**.

Generate and review the migration status report

Use the **t2c_report_extract** utility with the **-function-status_report** argument to generate a status report on the migrated data.

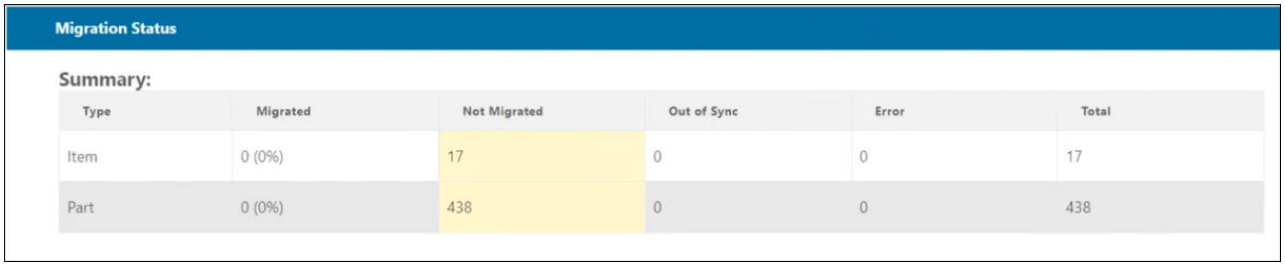
1. Open a Teamcenter command window on the source site.

2. Generate a migration report with the **t2c_report_extract** utility as in the following example.

```
t2c_report_extract -f=status_report -dir=d:\export\report01
-class=Item,Part
```

   This example reports only on the objects specified by **-class**. Exclude **-class** to report on all migrated objects. See **t2c_report_extract** for additional reporting options.

3. The migration status report files are created in the directory specified by the **-dir** argument. In that directory, open the *index.html* file in a browser to view the migration status report, for example:



   Note the following items when viewing the report.

   - The **Migrated** column shows the number of objects by type that have been migrated since the last time objects were registered using the **data_share** utility.

   - The **Not Migrated** column shows the number of objects by type that have yet to be migrated since the last time objects were registered.

- The **Out of Sync** column shows the number of objects by type that have been updated on the source site since they were migrated. Click on the reported number of objects to view a detailed list of the objects.

  When generating a report that contains out-of-sync objects, an *\OutOfSync* directory is created in the directory specified by the **-dir**. In that directory, *.txt* files are created for each type having out-of-sync objects. Consider using these files with the **data_share** utility to re-migrate only the out-of-sync objects.

- The **Error** column shows the number of errors encountered when migrating objects. Click on the reported number of errors to view a detailed list of the objects encountering errors when migrating.

  When generating a report that contains errors, an *\Error* directory is created in the directory specified by the **-dir**. In that directory, *.txt* files are created for each type generating errors. Consider using these files with the **data_share** utility to re-migrate those items after the errors are addressed.

Identify object differences and attribute mismatches

1. Open a Teamcenter command window on the source site.

2. Use a command of the following form to extract the site's objects as an island of data and save the data to a *.csv* file.

   ```
   t2c_report_extract -f=extract
   -output_file=d:\source_site\source_data.csv
   ```

3. Perform the same data extraction on the target site. Copy the resulting directory and .csv file to the same machine as the one with the source site extracted data.

4. Use a command of the following form to compare the contents of each site's extracted data.

   ```
   t2c_report_comparator -source=d:\source_site\source_data.csv
   -target=d:\target_site\target_data.csv -report_folder=d:\report
   ```

   The utility generates reports identifying missing objects, extra objects, and attribute mismatches between the source and target objects

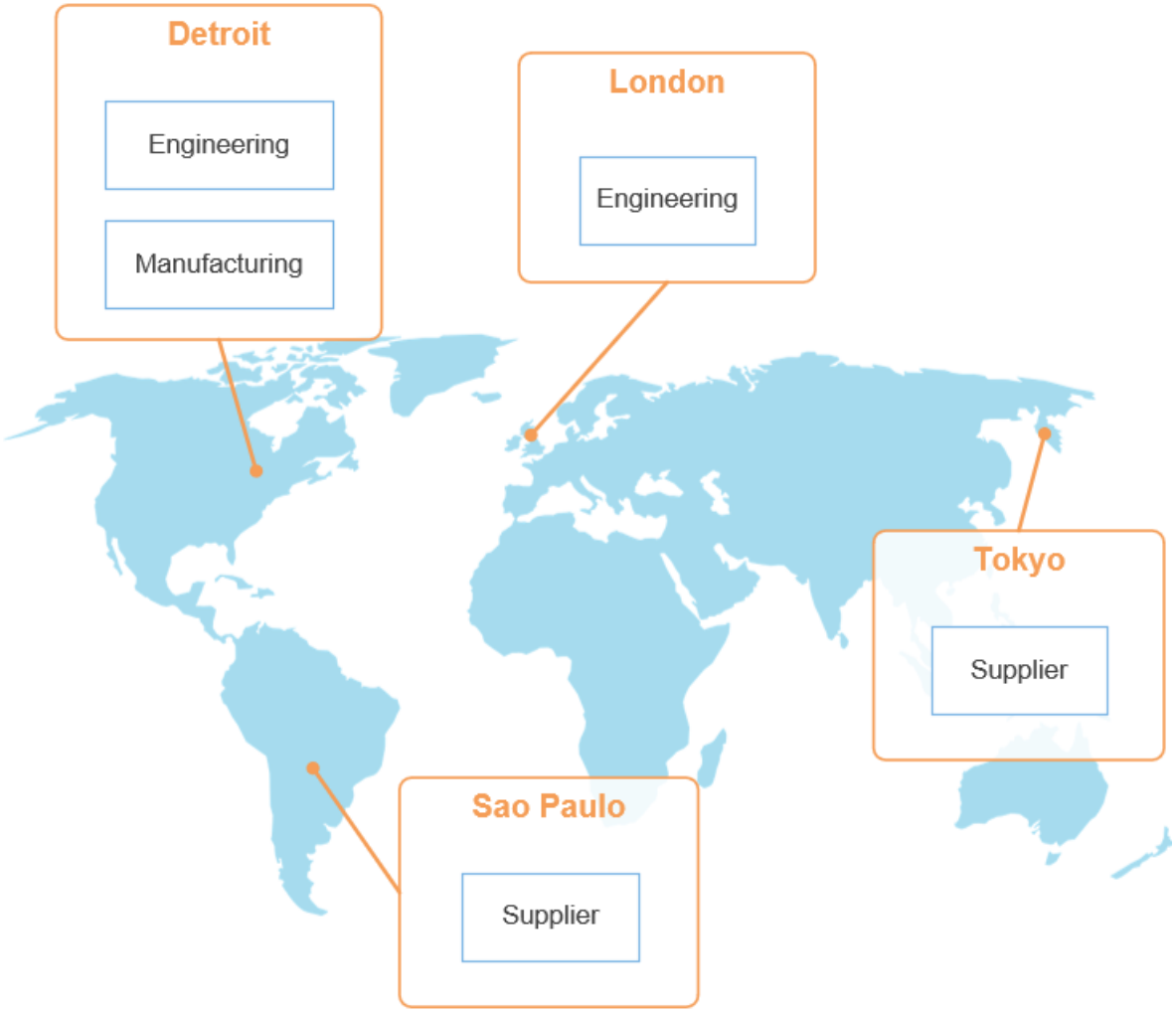
5. Review the reports generated in the directory specified by **-report_folder** for object and attribute differences.

5. Multi-Site basic concepts

Sharing product data across an enterprise

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

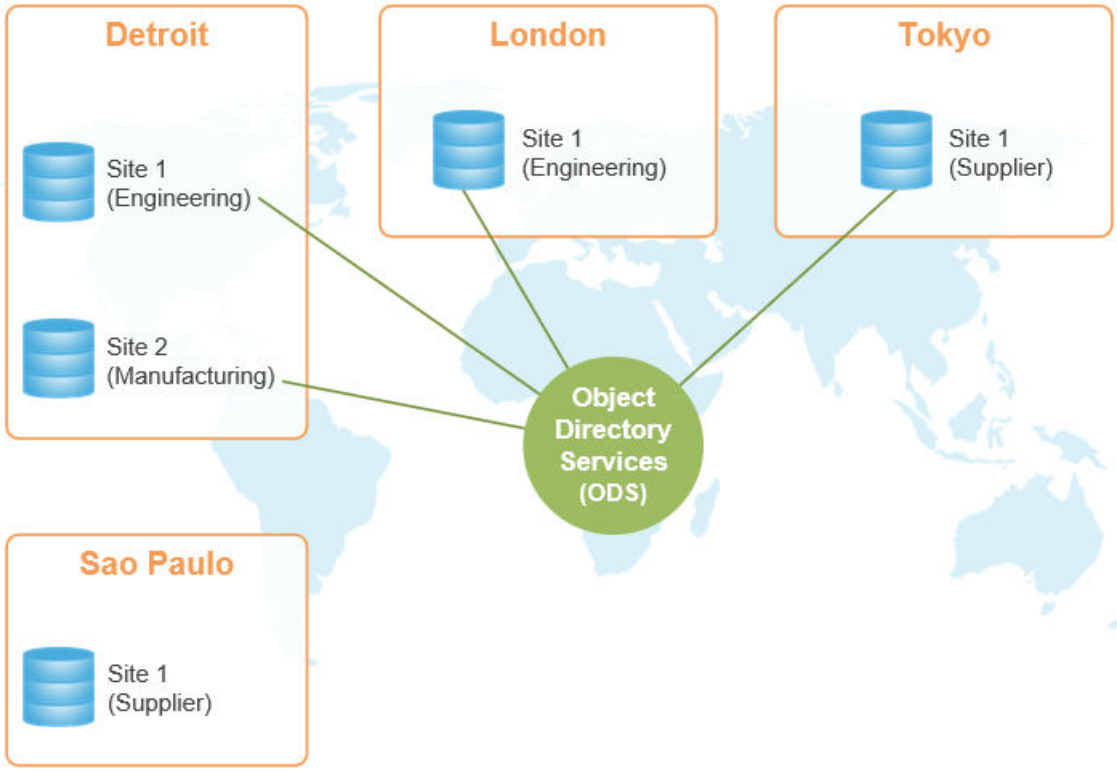
Protocol (FTP) or removable media (DAT) to the desired site, and manually importing them into the databases.

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.



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.

Term	Definition
Site	A single Teamcenter 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 for 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.
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 primary object can be replicated. You cannot replicate *replicas*. When an object is initially created and saved in a database, that instance is considered the primary 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 primary object can be modified. All replicas of the primary object are read-only. This ensures that the primary 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 primary object. • When transferring ownership to another site, only one site can be specified. This ensures that there is only one primary object in the network.

Object	Consideration
	• After it is replicated, a primary object cannot be deleted until all **replicas are deleted**. This ensures network-wide referential integrity.

Key concepts of TC XML-based data exchange

Following are key concepts for understanding how to use Multi-Site TC XML-based data exchange.

- **Closure rule**

  A closure rule controls the scope of the data translation for both import and export by:

  - Specifying how to traverse the data structure.
  - Defining the relationships that are of interest.
  - Defining what action to take when relationships of interest are encountered.

  Closure rules provide an efficient and codeless approach to extension of the data model. By default, Multi-Site uses the **MultiSiteDefaultCR** closure rule. For 4GD object replication or ownership transfers, the closure rules determine the contents of the island of data used in the transfer process. Contact your Siemens Digital Industries Software representative for information on using Multi-Site with 4GD data.

  Multi-Site closure rules and their controlling transfer option sets are used to provide direction to the **data_share** and **data_sync** utilities about how to traverse the Teamcenter database and how to process the information traversed. Teamcenter provides a set of default closure rules as part of the installation and upgrade process.

- **Dependent object**

  A dependent object that must be included in the set of objects implied by the primary object and the closure rules to correctly form the island of data. Not all primary objects have dependent objects.

- **Primary object**

  A primary object is the initial object that can be used to start the traversal to determine an island of data.

- **Island of data**

An island of data is a fundamental unit of transfer for moving data objects between sites using TC XML formatted data. It consists of a primary object and the additional objects on which it depends (identified by closure rules) for its correct functional definition and usage within Teamcenter.

For 4th Generation Design data, an island of data for reference transfer (replication) is different from an island of data for ownership transfer.

- **Scoper**

  When the exporter or importer receives a request to transfer data, it determines the transfer mode from the option set object. It then creates the traversal object based on the transfer mode scope (import, export) and the schema format (PLM XML or TC XML). The basic behavior of a traversal object is to provide two methods, **process** and **traverse**. The **process** method processes the present object in hand and calls the **traverse** method. The **traverse** method navigates to the next object and then calls the **process** method. This operation iterates until no more objects are available to process.

  The scoper evaluates the transfer mode closure rules and the options provided by the exporter and returns only the objects that are identified for the **process** method in a closure rule clause. This list is sent to the exporter or importer for serialization or deserializaton.

- **Synchronization**

  When a primary object is replicated at other sites, you must update the replicas whenever the primary object is modified. The process of updating replicas is referred to as synchronization. The synchronizer is responsible for ensuring that a previously replicated data is synchronized.

  In a Multi-Site environment, you perform manual synchronization using the **data_sync** utility. It is a re-export or re-import of the object of the object with the same transfer formula.

  > Caution:
  >
  > When manually synchronizing a replica, both the owning site and replica site must be online to receive replica deletion notification.

  > Note:
  >
  > TC XML-based data exchange does not support automatic pull synchronization.

  TC XML-based data exchange provides a traversal-free synchronization that provides better performance.

- **Transfer mode**

  A transfer mode () is a logical grouping of closure rule clauses. An administrator selects a transfer mode when creating closure rule clauses. Transfer modes allow users to export and import data by knowing only the transfer mode name that they must use, for example, **ToSiteA** or **FromSiteB**.

- **Transfer option set**

  A transfer option set (TOS) () is a stored set of transfer options used for remote data export. A transfer option set displays all of the unique options in the closure rule conditional clauses for the selected transfer mode. By default, Multi-Site uses the **MultiSiteOptSet** transfer option set.

- **Global workflow**

  TC XML-based data exchange supports workflow tasks across sites in your Multi-Site environment. You must configure remote inbox functionality and Security Services for this feature.

- **Central site**

  A central site is a site that contains all 4G Designer (4GD) data belonging to a Multi-Site Collaboration federation. Using a central site is a best practice for sharing 4GD data in a Multi-Site Collaboration environment. The data may be local or replica. Replicas on the central site are always kept up-to-date. Other sites use the central site as the comparison target site when using the **sync_on_demand** utility to determine synchronization status. Because a central site is a fully participating authoring site and owner of a program's partition structure, any site in the federation may be designated the central site except for a hub site.

Data synchronization

Replica-based synchronization

A replication-based solution must ensure that replicas are kept up-to-date when the primary object is modified. The process of keeping shared data up-to-date is called synchronization. You can synchronize data between sites manually or automatically. Multi-Site Collaboration maintains export records and provides synchronization tools you can use to keep replicas up-to-date.

Automatic synchronization allows Teamcenter to update replicas at remote site when a change occurs to the primary object at the owning site. This results in an efficient and evenly distributed synchronization process, and replicas are updated within minutes after the primary copy is modified.

There are important factors to consider when planning data synchronization. You must review **pull versus push strategy** for planning information regarding data synchronization to determine which types of synchronization you should use.

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 primary object. For items, a special

Item	Description
	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 primary 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.
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.
Automatic synchronization	The end user who replicates an object may specify that the replica be synchronized automatically when the primary object is modified. The replica is then synchronized automatically using the Multi-Site Collaboration automatic synchronization functionality.

Traversal-free synchronization

Multi-Site synchronization requires a full traversal to identify the modified objects and send those modified objects to the remote site, for example when the following structure is replicated to the remote site.

Assembly1
- **childComp1**
- **childComp2**
  - **childComp3**

If only the **childComp2** component is modified, full traversal synchronization must revisit all structures, starting from the **Assembly1** assembly to the final component in the structure (**childComp**...*N*) to identify that only **childComp2** is modified.

Traversal-free synchronization does not require traversal of all structures. Instead, SQL queries are used to quickly identify that only the **childComp2** component is modified. This same method also quickly identifies any added or deleted object, such as if you added a **childComp4** component or deleted the **childComp1** component in the **Assembly1** structure. Because the time to perform synchronization is proportional to the number of modified objects instead of the size of the whole structure, this provides much better synchronization performance.

Traversal-free synchronization is based on a *transaction*. The following are prerequisites for a transaction:

- For synchronization, the root objects must be the same as for the initial replication.

- The parameters used to collect objects must be the same as used for the initial replication. Therefore, the transfer option set must be identical for synchronization and the initial replication.

  For example, during the first replication, if you use the **-latest_revision** argument, you cannot use the **-all_revisions** argument for traversal-free synchronization.

The following process shows the use of traversal-free synchronization for the **Assembly1** structure:

1. Create a transaction during replication:

   ```
   Data_share -item_id=Assembly1 -include_bom -all_revisions
   -optionset -trid=AutoDesign001  -site=site2
   ```

2. Modify the **childComp2** object, add a **ChildComp3** object, and delete the **childComp1** object at owning site (**site1**).

3. Run traversal-free synchronization:

   ```
   Data_sync -trid=AutoDesign001 -optionset
   -site=site2 -sync -update
   ```

You can manage transactions for traversal-free synchronization using these additional parameters:

- To list all transactions:

  ```
  data_share -list_transactions -optionset
  ```

  or

  ```
  data_sync -list_transactions -optionset
  ```

- To clean up transactions:

  ```
  data_sync -cleanup_transaction -optionset
  -before_last_sync_date=specific_date
  ```

  or

  ```
  data_share -cleanup_transaction -optionset
  -before_last_sync_date=specific_date
  ```

- To generate synchronization report only:

  ```
  data_sync -sync -trid=AutoDesign001 -optionset  -site=site2
  ```

Synchronization utility preferences

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

The behavior of the **data_sync** and **data_share** utilities for project relationships on replica objects can be controlled by the **TC_sync_projects_with_owning_site** preference.

Use the **ADA_override_on_import** preference to specify whether or not a remote site mirrors the license-related properties of the owning site for a workspace object, for example, whether the Authorized Data Access (ADA) license attached to or detached from an item revision is propagated to the remote site.

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 primary 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 primary object is modified. This results in an efficient and evenly distributed synchronization process in which replicas are updated minutes after the primary copy is modified.

Additionally, you can request to be notified when the primary object of your replica is modified by selecting the **Notify by E-mail** option.

For complete option descriptions and requirements, see *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 primary 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.

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.

> Note:
>
> You may need to adjust the **File Locate** preferences in Teamcenter Lifecycle Visualization for session files created in a Multi-Site environment to load correctly.

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.

Using multiple attributes for object keys (unique IDs)

Multi-Site supports multifield key definitions for objects that it transfers. *Multifield* keys are identifiers assigned to each object to ensure their uniqueness in the database. Teamcenter administrators can add multiple properties to define a key. The multifield key is composed of a domain name (the name of the business object type) and a combination of the object's properties. This ensures a unique identifier across all the objects in the domain. You configure multifield keys for objects using the Business Modeler IDE.

Publishing and unpublishing objects

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.
data_share utility	Publish or unpublish objects in a batch using this utility.

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 primary object resides. It is the only site where the object can be modified or where you can obtain a replicated copy of the primary 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 primary 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 primary 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** this preference is defined and set to **TRUE**, and the owning group is defined at the importing site, the original owning group is preserved.

  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 Digital Industries 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.

Global organization objects

You can define organizational objects (**Group**, **Role**, **User**, **Person**, and **GroupMember** class objects) and then replicate them throughout your Multi-Site environment to provide a global organization. A global organization allows centralized administration for organization data using Multi-Site Collaboration technology. You select a central site in your environment to store the primary copies of organization data and the other sites contain read-only replicas. You can make changes at the owning site and synchronize the replicas at the other sites. Global organization objects can be imported and exported much the same as other Teamcenter objects, including transfer of ownership from the owning site to another site.

In an existing Multi-Site environment, sites may contain identical organization structures (cloned organization objects). These must be **migrated** to the replicated model to establish a true global organization.

To support a global organization, the **User** class contains a home site (**home_site**) attribute. This attribute defines the working site for the object which is the site where the user physically works. Users

that work at the owning site for the global organization objects may have a null value for this attribute. The home site attribute is used to determine the working site of a user or group in a global workflow. Based on this attribute an assigned task is delivered to a remote user at his or her working site.

The **User** class object also has a **remote_sites_deny_login** attribute. Users cannot interactively log on to sites listed in this attribute. However, users are always permitted to log on to their home site.

A primary object cannot be deleted if an export record exists for the object indicating the object has been exported (replicated at another site). Replica organization objects cannot be modified, therefore if an export record exists for an organization object it cannot be modified. For objects that have been migrated, no export record exists, however all objects that can be replicated have an **owning_site** attribute. This attribute prevents modification of replicas for migrated objects. To change replicated organization objects, you must modify the primary object and synchronize the replicas to pick up the changes. You can synchronize organization objects using the **data_sync** utility or by re-exporting the object to the remote sites from the Organization application. Synchronization is based on the last modified date of the selected objects.

You can also export organization objects from the Organization application by selecting the object and choosing **Tools→Export→Remote Export**.

The following preferences are related to replicating organization objects:

- **IDSM_global_dsa_sites_permitted_to_push_admin_data**
- **IDSM_global_dsa_set_local_volume_on_import**
- **IDSM_global_dsa_set_volume_on_import**

Organization objects replication behavior depends on the object being replicated:

Person objects are able to exist alone without related objects. Therefore, no traversal to related objects is performed.

Role objects own their **GroupMember** objects. When role ownership is transferred, all of its **GroupMember** objects are transferred. It is not possible to add a **User** object to a replica **Role** object as a result of this coupling.

The database administrator (DBA) group and DBA role cannot be exported through Multi-Site Collaboration or migrated using the **migrate_organization** utility.

Users can be assigned to project teams in conjunction with existing Teamcenter organizational roles. In a global organization environment, you cannot export or migrate the project team assignments.

User objects are always exported with their default **GroupMember** object. There is an option to send all of its **GroupMembers** objects. The group and role for the **GroupMember** object must already exist at the target site, otherwise the export fails.

Siemens Digital Industries Software recommends that sites deploying global organization should maintain one central site (the owning site). Make changes at the owning site and then push them to remote sites. If data must be modified at a remote site, temporarily transfer ownership to the remote site, modify the data at the remote site, and then transfer ownership back to the owning site.

Requirements objects

Multi-Site supports performing actions on requirement content in the same manner as other Teamcenter objects. You can:

- Import or export requirement objects and the related dataset content (named reference) with ownership transferred to the target site or as references (replicas) at the target site.
- Synchronize a replicated requirement object and its related dataset content.
- Remotely check out, modify, and check in requirement objects.
- Use the default **TIEUnconfiguredExportDefault** transfer mode for export and **TIEImportDefault** transfer mode for import of requirements objects.

If you have custom notes on your trace links, the custom notes are exportable using the **REQ_export_notesonlinks** transfer mode. Custom notes cannot be imported using PLM XML.

Exporting Systems Engineering Visio data

You can transfer Systems Engineering Visio diagrams using Multi-Site Collaboration. The image and dataset objects associated with the diagram are exported along with any relationship objects defined for Visio diagrams in the **TC_relation_required_on_export** preference. You can export diagrams for reference (replicas) or with ownership transfer. However, for ownership transfers, only the latest revision is transferred.

The following diagram objects and relations are transferred:

- Image dataset with attaches relationship (**TC_Attaches**)
- Visio (**Fnd0Visio**) dataset with attaches relationship (**TC_Attaches**)
- **POM_object** class objects with shape relationship (**Fnd0ShapeRelation**)

The following caveats apply to exported diagrams:

- Diagram templates are not transferred during diagram exports. You must export the associated diagram template separately.
- A diagram with stub objects or one that is not the root structure of the export cannot be opened at the importing site.

Multi-Site transaction logging

Teamcenter logs Multi-Site import and export transactions. The logs include export and import object counts, the time taken for each operation, and a **correlation ID** for each request in the administrator's **syslog** file. You can **set the priority of Multi-Site log messages** to the standard Teamcenter levels (**FATAL**, **ERROR**, **WARN**, **INFO**, **DEBUG**, and **TRACE**).

Part II: Configuring and administering Multi-Site Collaboration

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. The information in this section is intended to provide some practical guidelines that you, as a system administrator, can use to plan your Multi-Site Collaboration network.

6. Planning and setup

Audience

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 Digital Industries Software strongly recommends that anyone setting up a Multi-Site Collaboration network thoroughly review all planning considerations before performing any setup procedures.

Multi-Site Collaboration deployment options

Multi-Site is used to connect sites in several configurations:

- Multiple on-premise sites
- Multiple sites using cross-region cloud services
- On-premise sites and private cloud sites
- On-premise Teamcenter installations and TEAMCENTER X

When deploying Multi-Site Collaboration, choose from the following three deployment approaches:

RPC

Multi-Site data exchange uses the Open Network Computing Remote Procedure Call (ONC RPC) standard by default. Teamcenter IDSM and ODS RPC services are deployed in this approach. Be aware of the following items when considering this approach:

- The RPC approach is the default deployment configuration, supported when installing Multi-Site Collaboration using Teamcenter Environment Manager or Deployment Center.
- The ONC RPC standard has no built in encryption. Using the RPC approach requires that you separately configure VPN technology between sites in the Multi-Site federation for security.
- When using the RPC approach, you can disable the portmapper service on TCP 111 as described in **Bypass portmapper service**.

See **Multi-Site deployment-related preferences** for a summary of the preferences required to be set to use RPC mode.

Multi-Site using HTTP

This HTTP approach replaces ONC RPC with the Teamcenter web tier. The IDSM and ODS are replaced by **TcServer** processes started by the server pool manager. Authentication is managed by a shared Teamcenter Security Services (TcSS) domain.

Be aware of the following items when considering this approach:

- The approach provides enhanced HTTPS security using the Teamcenter web tier.
- TcSS authentication keys must be shared between sites in the Multi-Site federation.

See **Configure Multi-Site authentication using HTTP/HTTPS** for configuration details.

See **Multi-Site deployment-related preferences** for a summary of the preferences required to be set to use HTTP and a common SSO domain.

Multi-Site using secondary LDAP servers configured with TcSS

This HTTP approach uses Teamcenter Security Services configured with separate LDAP servers for authentication. A Multi-Site proxy Teamcenter user is created on each site and authentication is processed by separate Teamcenter Security Services SSO LDAP services using user names and passwords.

Be aware of the following items when considering this approach:

- The approach provides enhanced HTTPS security using the Teamcenter web tier.
- User name and encrypted password credentials must be managed for every remote site on each client site.

  For example, for Site1 to connect to Site2, Site2 is required to have a Site1 proxy user. This proxy user exists in Site2's Teamcenter database and secondary LDAP. Site1 does not need to create this user in their Teamcenter database, but must maintain a password file and add the proxy user name and password file to the **TC_alternate_sso_proxy_table** preference.

See **Configure Multi-Site authentication using secondary LDAP servers configured with TcSS** for configuration details and an example workflow.

See **Multi-Site deployment-related preferences** for a summary of the preferences required to be set to use HTTP and separate SSO domains.

Advanced concepts

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 on to the ODS database using a Teamcenter administrator user account.

The IDSM daemon is dynamically started using the **run_tc_idsm** script and runs until it has accomplished its task of transporting a set of objects from one site to another. It then transitions to a dormant state for 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 Digital Industries 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.

Listing running RPC programs

To return a list of RPC programs running at the specified node, enter **rpcinfo -p** *node_name*. Results are displayed as follows:

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

If an error is returned or there are no results, 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
536875585	1	TCP	2021	Not applicable

Testing daemon readiness

To test whether a daemon is ready, enter **rpcinfo -T tcp** *node_name program_number version_number*. (On some platforms, the syntax of this command requires a lowercase **t** as in **rpcinfo -t**.)

If the daemon is ready, a message similar to the following is returned.

```
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.

Modifying shared data

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 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 primary 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 Digital Industries Software recommends using this method whenever possible.

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 primary copy. When the primary 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 primary copy. The information in the PAR is used to determine if an object needs to be republished, for example, 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 primary copy from being deleted, ensuring network-wide referential integrity. To delete a primary 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.

Planning and setup process

Siemens Digital Industries Software recommends the following sequence of tasks when setting up a Multi-Site Collaboration network.

1. 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.

2. 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.

3. 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.

4. 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.

5. 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

Determining how to share data

The optimum Multi-Site Collaboration configuration varies greatly from enterprise to enterprise. 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 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

Site category	Description
	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.

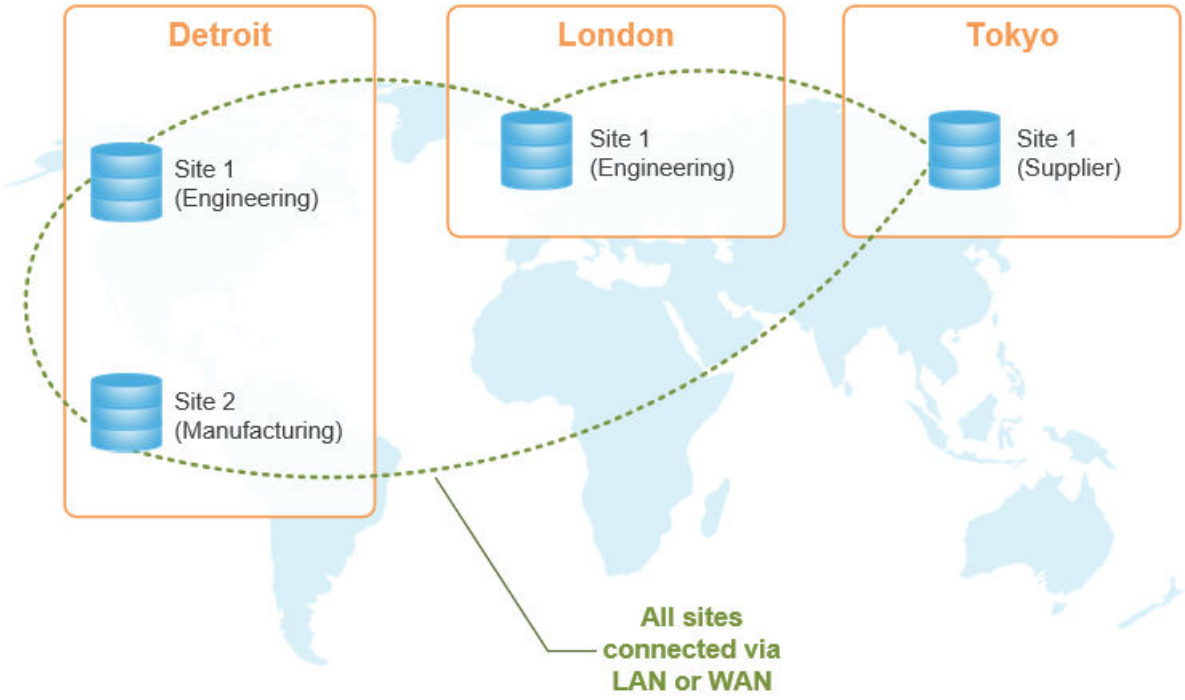
Planning your network

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 networks can have the following topologies. For internal company sites, Siemens Digital Industries Software recommends starting with a peer-to-peer topology so the sites can freely share data among themselves. Then, to solve specific problems, such as limiting supplier access, use a hierarchical topology by creating a supplier central library. Sites can be installed on premises or be **cloud-based**.

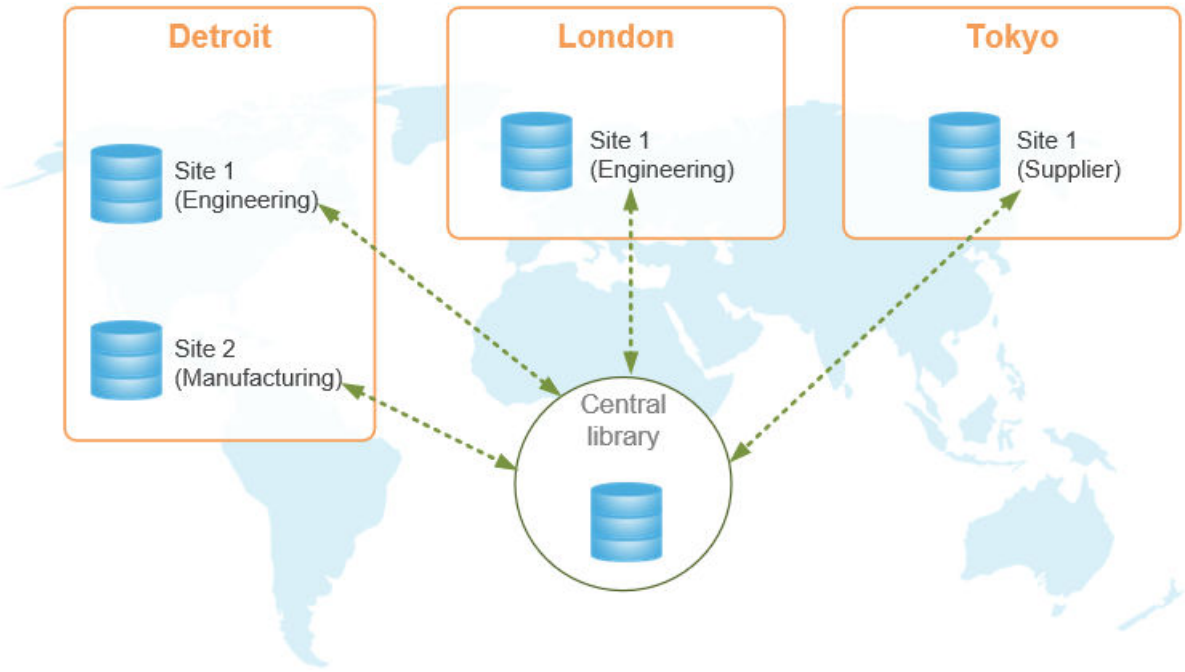
Peer-to-peer

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).



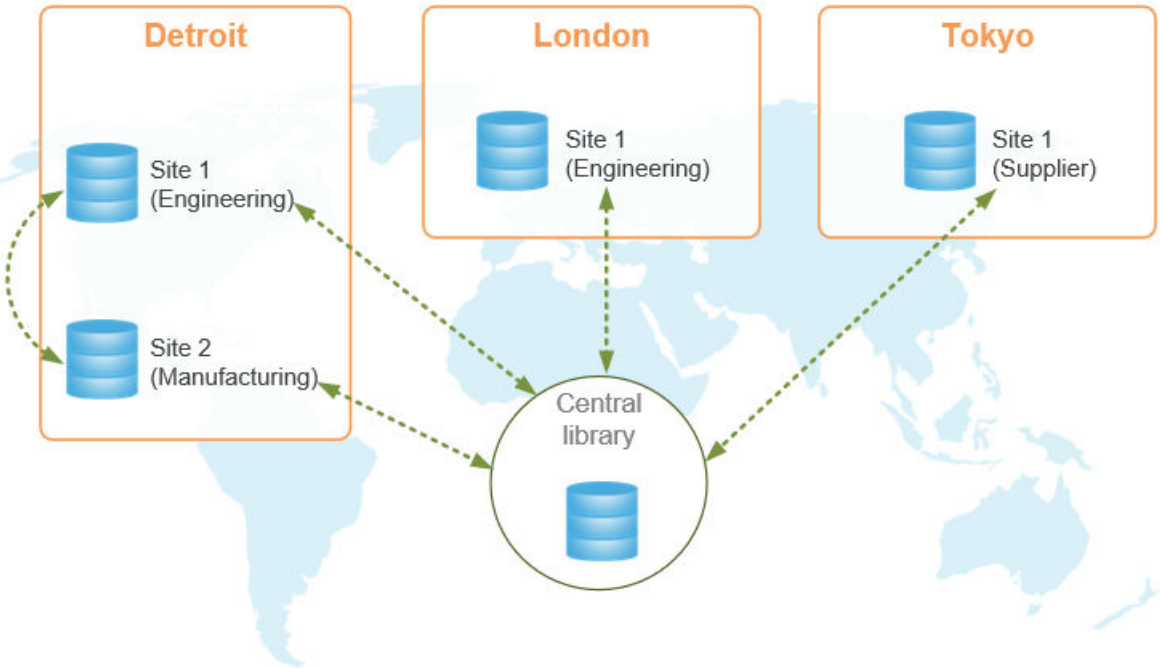
Hierarchical

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 primary objects used in the enterprise. Because all primary 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.



Combination of peer-to-peer and hierarchical

A Multi-Site Collaboration network can also be configured 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 and to the central library through a LAN or WAN.



Structured context object 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. 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 File Management System (FMS) server cache (FSC). This is most useful when you have knowledge of what data a remote site works with frequently, for example, 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, for example, 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 primary configuration 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** 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.

There may be times when you want to add remote files, for example, library parts such as common fasteners, to the volume instead of the cache. Because the **TC_force_remote_sites_exclude_files** and **TC_Populate_FSC_Server_Targets** preferences are set for **ALL** scope, you can set these values with environment variables of the same name on the local host. This allows importing library parts to replica volumes for that particular host. Use the **tc_profilevars** file to set these variables.

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.

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.

Connecting external sites using a hub site

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.

Caution:

By default, the import process defines any site that owns an imported part in the importing site's database if the site is not already defined. This may be unacceptable because it makes the owning site information visible to users at the importing site. To prevent automatic site definition, set the **TC_do_not_define_sites_on_import** preference at the importing site.

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.

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.

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.

Synchronization considerations

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 primary 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 no means of overriding these defaults. However, for the **data_sync** utility, it is possible to override these defaults.

Synchronizing hub site 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 site guidelines

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.

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

Criteria	Description
Geographical location	Siemens Digital Industries 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).

Determining 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.

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 process 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 workflow handlers

     Estimate the number of objects pushed by workflow 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.

Object directory services (ODS) sites

Object Directory Services 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.

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

Consider running the ODS in multiprocess mode if:

- Your ODS site is providing slow service due to a high number of ODS operations from remote sites.

- Remote users are performing time-consuming operations, for example, generating reports about published objects.

Number of Object Directory Services sites required

The optimum number of Object Directory Services (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 network bandwidth and server node requirements

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.

The ODS server node is the network node running the ODS daemon. Siemens Digital Industries 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.

Network considerations

Adjusting the network for your work patterns

After you determine your system and network configuration, you should make adjustments based on work patterns and time zone differentials. You must consider network activities 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.?*. Also, determine if the configuration needs adjustments for the following:

Total megabytes-per-day	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 several gigabytes per day, providing sufficient bandwidth between these sites is an important consideration. 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 (for example, 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** 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	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.
IDSM CPU and memory	The IDSM server node requires a heavy-duty CPU. For information about how CPU and memory options affect Multi-Site performance and throughput, see *Teamcenter Deployment Reference Architecture* in the Teamcenter downloads area on Support Center, under **Support White Papers Teamcenter Deployment Reference Architecture**.
IDSM hosting	Because an IDSM server is always associated with a single database, you may consider using the database node to host an IDSM server. This provides faster import and export operations 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 can slow down the overall system performance. Unless you are willing to upgrade the database server node, Siemens Digital Industries Software recommends not using it as the IDSM server node.

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 primary 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 Teamcenter administrator account 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 Digital Industries 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.

The method the Teamcenter installation process uses to generate site IDs required an upgrade for IDs generated beginning on January 1, 2010. Prior to this date, the **generate_site_id** utility generated only positive integer site IDs. The available set of IDs was exhausted, requiring a utility upgrade to generate negative integers.

If you have not downloaded the upgraded utility, you cannot create a new database. See the **Teamcenter_site_id_utility_DetailedDoc** document available from the Teamcenter download page in Support Center for information about the upgraded utility.

If you have sites with negative integer site IDs in your Multi-Site environment, you must ensure that any utility argument, configuration file entry, or dialog box entry contains the minus (-) sign prefix for the negative integer site IDs.

Multi-Site Collaboration license considerations

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.

Multi-Site Collaboration ACS licenses are of the following types:

- ODS license
- Distributed User license

ODS licenses

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 checked during all Multi-Site Collaboration ODS operations. Ensure you have purchased an ODS license for each ODS site you plan to run in the network. If you do not have the required ODS licenses, your Teamcenter users receive an error message when attempting any Multi-Site Collaboration operation. All licenses can be placed in one ACS accessed by all ODS processes or they can be distributed to several ACS sheets.

Distributed user licenses

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 *Teamcenter Environment Variables*.

Securing a site

Set up user-level security for a site by including the site in the **TC_check_remote_user_priv_from_sites** preference for the IDSM server.

Access Manager (AM) validation is performed for user requests from sites that are set in this preference and the following 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 engineering process management. However, the 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 and remote import with transfer of ownership
- Remote export and remote export with transfer of ownership
- Remote check out
- Data share utilities
- On-demand and pull 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 and remote import with transfer of ownership
- Remote export and remote export with transfer of ownership
- Remote checkout

Local Teamcenter administrator user privileges 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 user with Teamcenter administrator privileges	
Remote import with transfer of ownership	Remote user				Remote site and optional user with Teamcenter administrator privilege
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 primary 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 (one with Teamcenter administrator privileges) that is used to run the IDSM server.

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

Be aware of the following items when working with 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.

Controlling user access to remote sites

Control local users capabilities when accessing remote sites as follows.

- 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** 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 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** preference.

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

While your users can search other ODS sites, as defined in the **ODS_searchable_sites** 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** 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 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 *Requirements for Multi-Site system administrators*.

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 primary 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 primary 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 nonreplica files. You can segregate replica files using the **TC_replica_volume** preference to indicate the volume into which replica files are stored. If you set this preference group protection scope, you can set up 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 preference with group protection scope as:

TC_replica_volume = **eng_replica_volume**

A site protection scope 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 primary object.

Remote checkin/checkout control

Using preferences to control remote checkout access

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.
IDSM_permitted_checkout_users_ from_site_*sitename*	Defines which user IDs from the sites specified by the **IDSM_permitted_checkout_sites** 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 **IDSM_permitted_checkout_sites** preference may perform remote checkouts of objects owned by the local site.

Using access rules to control remote checkout access

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 access

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.

Controlling transfer of ownership

Control transfer of ownership using preferences and access rules.

Transfer of ownership preferences

The following two preferences are defined by default with the protection scope set to site. They control the ability of remote users to transfer site ownership of objects owned by your site.

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 **IDSM_permitted_transfer_sites** preference are authorized to transfer ownership of objects owned by the site served by an IDSM server.

Transfer of ownership 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.

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.

> Caution:
>
> Multi-Site Collaboration does not pass code set information with the metadata transferred between sites. Therefore, the sending and receiving host must use the same or compatible code sets in a Multi-Site environment. Some code sets are not supported by some host types. You must consider this when planning your Multi-Site environment.

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

- 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* preference must be set for each monolingual site at multilingual sites that export to the monolingual site to support publish and remote search operations.

  Additionally, if this preference is not set for a site, Multi-Site assumes the site is multilingual.
- 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.
- 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.
- When you use remote check in/check out to transfer objects, the check-in operation involves site ownership transfer. Therefore, if the transfer is from a multilingual site to a monolingual site, this also results in loss of localized values.
- 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.

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, Linux, 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.
# 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, Linux, 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 LOCATION PREFERENCE SETTINGS	
Site Name:	
Working (IDSM) Preferences	
IDSM_permitted_sites=	ODS_site=
IDSM_permitted_transfer_sites=	ODS_searchable_sites=
IDSM_restricted_sites=	ODS_searchable_sites_excluded=
	TC_publishable_classes=
TC_transfer_area=	
ODS Preferences	
ODS_permitted_sites=	ODS_restricted_sites=
TC_ods_client_def_timeout=	TC_ods_client_initial_timeout=
Language Preference	
TC_master_locale_*site-name*=	
Replica 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=

Setup procedures

Determine the setup process

Review the *Planning considerations* before beginning the setup procedures as some procedures are optional and some have optional configurations. The setup procedures may be dependent on other procedures. Follow this process to ensure that you have the required procedures completed before you begin a dependent procedure.

1. **Configure the sites** in your Multi-Site network.

2. Configure File Management System (FMS).

3. (Optional) Configure global data caching.

4. (Optional) Set up a hub site.

5. Create remote 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).

6. Synchronize 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.

7. (Optional) Configure multiprocess Object Directory Services (ODS).

8. (Optional) Set up partial item export.

Configure Multi-Site Collaboration sites

Initial setup of Multi-Site Collaboration sites is accomplished using the installation script. Installing Teamcenter is described in detail in the Teamcenter installation guides for Windows and Linux. Initial configuration of Multi-Site Collaboration is described in **Enabling Multi-Site Collaboration**.

The following steps briefly describe how to use the installation script to set up and complete the configuration of your Multi-Site Collaboration sites.

1. Install Teamcenter and **initially configure Multi-Site Collaboration**.

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 Multi-Site Collaboration with cloud-based Teamcenter installations

Multi-Site Collaboration can be deployed with Teamcenter installed in a virtual private cloud (VPC).

Cloud and network configuration details are specific to each installation. Once your organization is configured to use a VPC, install one or more cloud-based instances of Teamcenter on VPC virtual machines.

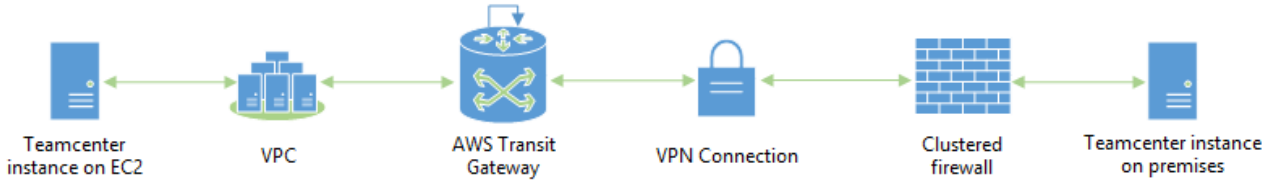
The prerequisites to using Multi-Site Collaboration with cloud-based instances of Teamcenter are as follows:

- Network communications must exist between the Teamcenter server environments.
- The TCP ports for ODS, IDSM, FSC, and ONC Portmapper must all be open. **ONC Portmapper can be optionally bypassed**.

The following examples illustrate cloud-enabled Multi-Site Collaboration configurations using Amazon Web Services Elastic Compute Cloud (EC2).

On-premises-to-cloud

In this configuration, Teamcenter instances run within a LAN while collaborating with cloud-based instances.



Cloud-to-cloud

In this configuration, Teamcenter instances running in separate cloud regions collaborate.

Configure FMS for Multi-Site Collaboration support

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 primary configuration file for each site. This file is located in the *TC_ROOT***fsc** directory for each Teamcenter site. The following examples show how to set these elements for two sites on the network:

FMS primary configuration file – site 1

```
<?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_tcdba" />
      </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 primary configuration file – site 2

```
<?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_tcdba"/>
      </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>
```

Configure FSC authentication for Multi-Site Collaboration

You can configure Multi-Site Collaboration to leverage File Management System (FMS) enhanced ticket authentication. With this configuration, Multi-Site uses Teamcenter Security Services (TCSS) to authenticate FMS tickets. Tickets are successfully authenticated only when the requester presenting the valid security ticket is the requester who generated the ticket.

Prerequisites

Ensure the following conditions are met before configuring Multi-Site Collaboration to communicate using FSC authentication.

- File Management System (FMS) enhanced ticket authentication is configured at your site.
- Teamcenter Security Services is installed and configured.
- Multi-Site Collaboration is installed and sites are communicating with each other (without FSC authentication configured).

Configuring FSC authentication for Multi-Site Collaboration

1. On each site, set the following preferences:
   - Set **TC_remote_sites_requires_auth_fsc_token_exchange** to the name of one or more remote sites which require authentication before importing volume files.
   - Set **TC_tms_auth_fsc_app_id** to the local site's Teamcenter Security Services application ID.
2. On each site, update the **multisiteimport** section of the FMS primary configuration file to use FSC authentication. Following is an example configuration. See FSC group import multisite routing configuration and the neighboring topics for details and additional options for configuring Multi-Site FSC communications.

   ```
   <multisiteimport siteid=your_site_id>
     <defaultfscimport fscid="fsc_id" fscaddress="http://target_IP:port" priority="0"/>
   ...
   </multisiteimport>
   ```

   Where:

   fsc_id

   The FSC ID of the remote site. That is, site1 sets this value to the FSC ID of site2.

   target_IP

   The IP address of the remote site.

   port

The listening port on the remote site.

For example, site1 is set as follows.

```
<multisiteimport siteid=-1681897519>
  <defaultfscimport fscid="fscsite2" fscaddress="http://
10.22.154.118:5502" priority="0"/>
...
  </multisiteimport>
```

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**.

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"/>
```

Set up a hub

A **hub configuration** allows the replication of replicas in a controlled manner.

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 Digital Industries Software recommends that the site name includes a prefix or suffix 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**.

To use a hub with HTTP or HTTPS protocol, the following are required:

- Teamcenter four-tier
- A lightweight directory access protocol (LDAP) or Windows Active Directory
- Single sign-on (Security Services)
- The **tcssoid** and **tcssols** WAR files created with Web Application Manager and deployed to a web server

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** preference. Each client site must be defined in this preference.

4. Define and assign values to the **IDSM_permitted_transfer_sites** 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** preference to include all ODS sites accessible to the hub, including the hub ODS.

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

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

The hub client is configured.

Creating remote 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.

Distribute 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.

Configure a multiprocess ODS

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

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

2. Set the **ODS_multiprocess_initial_subprocess_count** 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**.

3. Set the **ODS_multiprocess_max_subprocess_count** 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.

Configuring an IDSM and ODS processes on an alternate server

Prerequisites for a stand-alone IDSM and ODS server

Typically, you install the IDSM and ODS server along with your Teamcenter corporate server using Deployment Center. Running the IDSM and ODS services on an alternate server is not a supported configuration for Deployment Center. However, you can create a custom configuration to host these processes on a separate server after you complete primary installation using Deployment Center.

The procedure provided describes one possible way to perform this customization. Your network topology, firewalls, and security policy can complicate or even prevent this configuration. These issues are not in the scope of Teamcenter help.

The process requires you have the following skills:

- Ability to install, configure, and test the IDSM and ODS server processes on the corporate server.

  > Caution:
  >
  > You must ensure the IDSM and ODS server processes are installed, configured, tested and running properly on the corporate server prior to installing them on the alternate server.

- Ability to create a network share for the *TC_ROOT* and *TC_DATA* file systems from the corporate server to the alternate server.

- Ability to create and maintain Windows services or Linux startup daemon processes.

Configure IDSM and ODS processes on a separate server

1. Using Deployment Center on the alternate server, install only the Multi-Site Collaboration Proxy Server solution. Do not select any other features or solutions. This installs the IDSM and ODS services.

2. Locate the **run_tc_idsm** and **run_tc_ods** scripts in the **bin** directory of the Teamcenter installation. These are shell scripts on Linux systems and batch files on Windows systems.

   Make the following changes to these scripts:

   a. Change the value of the **TC_ROOT** entry in the scripts to point to the network shared version of the *TC_ROOT* directory on your corporate server. You must provide a UNC path on Windows systems.

   b. Change the value of the **TC_DATA** entry in the scripts to point to the network shared version of the *TC_DATA* directory on your corporate server. You must provide a UNC path on Windows systems.

   c. Remove all environment variable settings that pertain to the **TC_PREFERENCES_OVERRIDE_FROM_FILE** preference or shared memory.

   d. Restart the ODS and IDSM server processes using the steps appropriate for your site.

3. At participating remote sites, launch the Teamcenter rich client and open the Organization application. Change the **Site Node/URL** box to point to the alternate server instead of the original corporate server.

4. Test the configuration.

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:

- Export selected revisions only
- Exclude export protected components of an assembly

Both of these options enable the owning site to control which parts of an item or assembly 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 use accessors to protect objects from import and export, Siemens Digital Industries 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.

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, then the selector that was 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.

Setting up automatic synchronization

Automatic synchronization preferences and events

The **TC_sync_auto_synchronize** preference lets a remote site specify that replicas are **automatically synchronized** when the primary objects are modified. The **Synchronize automatically** check box in the **Remote Export Options** and **Remote Import Options** dialog boxes indicates when automatic synchronization is enabled. This check box is informational and read-only. If the **TC_sync_auto_synchronize** preference is set to **TRUE**, the **Synchronize automatically** check box is selected and Teamcenter performs automatic replica synchronization.

Automatic synchronization requires that you set the **TC_subscription** preference value to **ON** at the owning site. This preference enables the subscription functionality that automatic synchronization requires.

You can set up subscriptions using the following events that affect your Multi-Site environment.

Event type	Description
__Replica_Update	Notification of a replica update when you manually subscribe to the event at the replica site. This event is triggered upon modification of a replica. Requires the **TC_subscribable_replica_classes** preference value be set.
__Register_Failed	Notification of failure to register an object.
__Unregister_Failed	Notification of failure to unregister an object.
__Publish_Failed	Notification of failure to publish an object.
__Unpublish_Failed	Notification of failure to unpublish an object.
__Remote_Check_In	Notification of a remote checkin of an object.
__Remote_Check_Out	Notification of a remote checkout of an object.
__Remote_Cancel_Check_Out	Notification of cancellation of remote checkout of an object.
__Remote_Transfer_Check_Out	Notification of a remote transfer and checkout of an object.

Automatic site synchronization 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.

You can also access the help information by running the daemons as a utility with the **-help** argument.

Synchronization email notifications

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 site's database.

  You can also access the help information by running the daemons as a utility with the **-help** argument.

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

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

Synchronizing objects

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.

Your enterprise must choose a strategy that is appropriate for the way it does business. The multiple site synchronization strategy is suitable for sites that have 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. You should you run the **data_sync** utility 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

In some situations, a strategy of synchronizing only modified objects may not result in optimum efficiency because the modified-only option involves some preprocessing. Teamcenter 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 to which 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 more efficient operation. However, as the number of sites being synchronized increases, the number of IXRs 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** argument to override the modified-only option.

Therefore, do not use the **-disable_modified_only** argument except when analyzing efficiency problems associated with the **data_sync** utility.

Projects

When you add or remove an object from a project, the object itself is (typically) not modified. Therefore, the last modified date of the object is not changed. This can cause the last exported date to be later than the last modification date. To track changes to the project assignment of an object, Teamcenter creates a project object relation (POR) and attaches it to the object. The POR contains a last modification date attribute that the **data_sync** utility checks against the last exported date to determine if the object requires synchronization due to a change in project assignment.

If you remove an object from the only project that it is assigned to, the POR for the object is removed. To track this type of change, Teamcenter modifies the last exported date of the IXR to a year that predates Multi-Site Collaboration functionality. Also the last modified date and last modified user on the object are updated. This causes the **data_sync** utility to indicate the object requires synchronization regardless of the last modification date of the object.

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** 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 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 (Object Directory Services) 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.

The Multi-Site Collaboration 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.

7. System administration

Requirements for Multi-Site system administrators

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)

Distributed environment considerations

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 Digital Industries Software recommends using a corporate-wide naming convention for temporary IDs, just as Siemens Digital Industries 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 Digital Industries 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

Protecting shared data using Access Manager 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 primary 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.

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 Digital Industries 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**.

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** preference to **TRUE**.

Controlling the remote import capability

Rule tree control of the 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.

To implement the more granular control you must:

- Identify the remote sites that you want to control the remote import capability for and create named ACLs for each 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 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 named ACL.

   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 remote site

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
```

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

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.

Transferring data among sites with different schemas

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.

Previous Teamcenter versions use a **$** character in the file name. The current version uses a **-** character in its place. To make POM transmit schema files compatible with among versions of Teamcenter and Multi-Site Collaboration, you must set the POM transmit **POM_TRANSMIT_OLD _NAMES** variable on sites that use the **-** character and the **POM_TRANSMIT_NEW _NAMES** variable on sites that use the **$** character. Make sure that the variables are set opposite of one another at the sites to ensure that they generate and locate the correct file names.

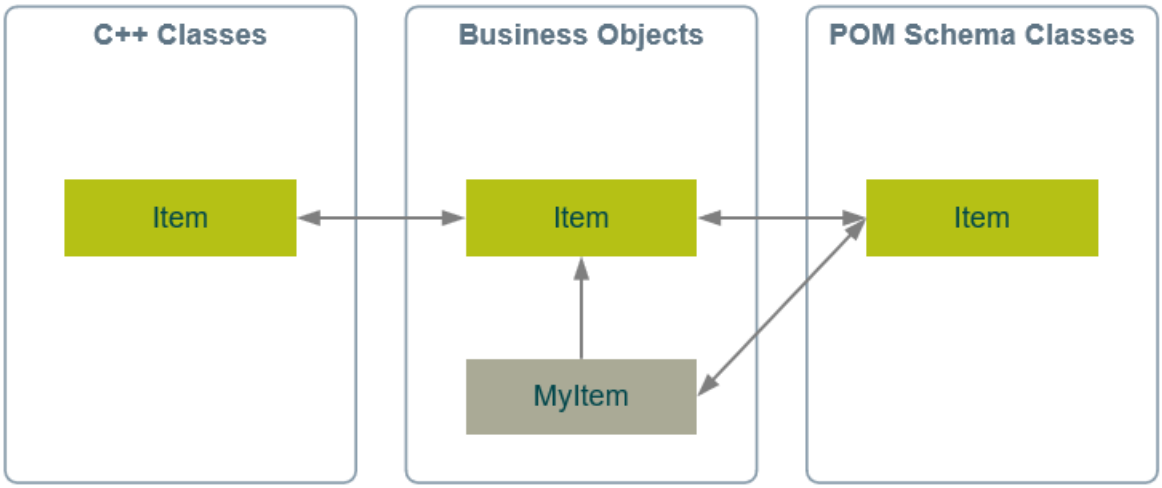
To configure a site that is running a current version of Multi-Site Collaboration, you may generate a POM transmit schema file that is compatible with a previous version by setting the **POM_TRANSMIT_OLD _NAMES** variable. If you set the variable to **ON**, the POM transmit schema file is created with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

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.

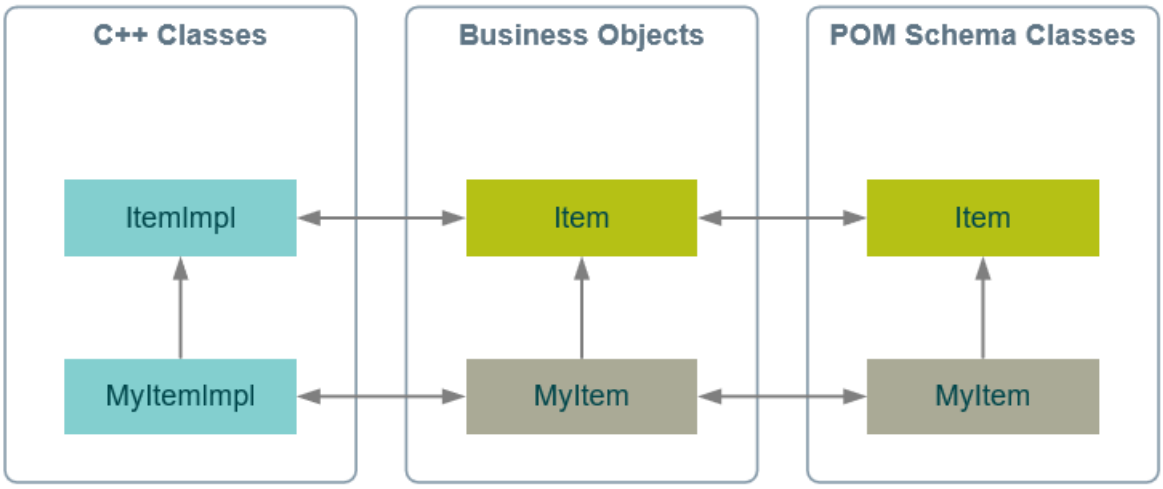
Backward compatibility of extended attributes

Any custom attributes on forms in earlier Teamcenter versions are not imported into the primary class object in the current version of Teamcenter. 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.



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



Any additional attributes on a subtype primary 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.

You can convert secondary objects to primary objects to avoid the loss of secondary object attributes.

You can add attributes to forms in Teamcenter 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

> Note:
>
> This information applies to allowing data transfers between the current version of Teamcenter and versions earlier than 11.0.

You must create a dataset mapping file to allow data transfers with earlier versions of Teamcenter. 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.

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.

- 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

Set the **Transfer out** privilege for the IDSM user at the remote sites.

Caution:

If you add arrangement relationships to the required relation preferences, you may encounter a case where you cannot export a subcomponent of a structure. This occurs because one of the subcomponent arrangement objects was transferred when the parent component was transferred to another site.

By default, arrangement data for each component level can be shared by explicitly selecting the relationship for transfer. This is the preferred method to avoid issues with setting arrangement relationship values in the **TC_relation_required_on_transfer** or **TC_relation_required_on_export** preferences.

Consolidating duplicate item IDs

The item ID process flow and Multi-Site-related 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 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 list of parts that the site owns is the site's primary data. The parts that the site does not own are duplicates that are replaced by replicas.

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

   The primary 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.

Process	Action
	• Access the Multi-Site Collaboration network environment and use the **Remote Import** command to import the primary items from the owning site. • Import all the NX items that are identified as primary 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.
Manual process	Delete the duplicates data objects.

Removing a site from a Multi-Site environment

There are many reasons you may want to remove a site from a Multi-Site environment, such as infrastructure cost reduction, Information Technology (IT) centralization, and business acquisition or mergers. In almost all cases, you must ensure that the data at the site being removed is not lost at your remaining Teamcenter sites in the environment and to ensure the data integrity at the remaining sites.

You use the site consolidation process to accomplish the removal and avoid loss of data or data integrity. The site consolidation process requires a multidisciplinary team versed in the business, legal, infrastructure, data management, and end-user issues impacted by the effort.

Enabling archiving and restoring with Multi-Site Collaboration

Prerequisites

Ensure the following conditions exist before enabling archiving and restoring.

- Each production and archive site must have Teamcenter 11.4 or later installed. Siemens Digital Industries Software recommends all sites have the same version of Teamcenter installed.
- Archiving requires the purchase of an additional license.
- Multi-Site Collaboration must be installed, and the owning and archive sites must be part of the same Multi-Site Collaboration federation.
- ADA licenses, projects, and security configurations must be consistent between the archive and owning sites.

- Production and archive databases must be of the same type. That is, an Oracle production database can only be connected to an Oracle archive database.
- When connecting a production database to an existing database as an archive, the two databases should be similarly structured.
- Closure rules related to **SiteConsolidationDefault** and **SiteConsolidationLW** must be the same between production and archive sites.
- To maintain traceability, users and organizations on an archive site should match those on the owning site. Objects ownership and audit history will not be maintained if the archive site does not have the same users defined. If user and organization administrative data is not consistent between the archive and owning sites, standard Multi-Site Collaboration ownership rules as described in **Object ownership and protection** apply, with ownership transferring to the user owning the IDSM process.

Install, configure, and connect to a new archive database

The Teamcenter Environment Manager is used to connect to an existing database that will be designated as an archive database. TEM is also used to populate a new database to be used as an archive database. The following procedure covers both scenarios. (Depending on the type of connection you are creating, the order or number of TEM panels and their contents may vary from those in the following steps.)

1. (Windows) Launch TEM in maintenance mode. In the Windows start menu's **Teamcenter** group, right-click **Environment Manager** and choose **Run as administrator**. Alternatively, you can run the **tem.bat** file in the **install** directory in the application root directory for the Teamcenter installation. Right-click **tem.bat** and choose **Run as administrator**.

   (Linux) Start TEM in maintenance mode by changing to the **install** directory in the Teamcenter application root directory for your Teamcenter installation and running the **tem.sh** script.

2. In the **Maintenance** panel, choose **Configuration Manager**.

   For more information about any panel in TEM, click the help button  .

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

4. In the **Old Configuration** panel, select the configuration you want to modify.

5. In the **Feature Maintenance** panel, under **Teamcenter Foundation**, select **Connect Archive Database**.

6. In the **Archive: Create Connection** panel, select the type of connection you want to create.

   - **Connect to existing configuration**

Connects your configuration to another configuration at the same site, using one database for production and archived data. Select the configuration to set as the archive configuration.

- **Connect to existing database**

  Connects your configuration to an existing archive database at another site.

- **Populate archive database**

  Connects to and populates a new database at another site, designating that database as the archive. Optionally choose to create a database on the other site if one does not already exist.

7. In the **Teamcenter Administrative User** panel, enter the password for the Teamcenter administrator.

8. In the **Operating System User** panel, type the password for the system user. This user cannot be **infodba**, and must be a system administrator with dba privileges.

9. In the **Archive: File System Cache Service (FSC)** panel, type the required values for creating the FSC. The FSC ID must be unique and any available port value can be specified. Check **Enable configuration master** to make the archive's FSC a primary FSC.

10. In the **Archive: Foundation Database** panel, enter access information for the database.

    For **Data Directory**, enter a location for the Teamcenter data directory. The directory must exist. The Teamcenter data directory is called the *TC_DATA* directory.

    If you are creating a new database, also specify the system user credentials and the path for the new database.

11. In the **Archive: Volume Information** panel, enter a name for the volume to create and the absolute path to the directory in which to create the volume.

    Siemens Digital Industries Software recommends not defining the volume location under the Teamcenter application root directory. Doing so leads to complications when upgrading to a later version of Teamcenter.

12. In the **Archive: Foundation Settings** panel, specify the foundation settings for the archive database as follows.

Value	Description
Transient Volume Directories	Specifies transient volume locations for Windows hosts, Linux hosts, or both. A transient volume is an operating system directory controlled by Teamcenter and used to store temporary data for transport of reports, PLM XML

Value	Description
	data, and other nonvolume data between the enterprise tier and client tier in a deployed four-tier architecture. All four-tier clients that access the corporate server you are installing use this transient volume. Caution: You cannot define the path as a UNC path, for example, *****server******shared-transient-folder* . You must use a direct path location. This is partly due to the fact that some ZIP archive utilities do not accept UNC paths, resulting in failure of exports to Excel or Word.
Windows clients	Specifies the location for a transient volume for Windows client hosts.
Linux clients	Specifies the location for a transient volume for Linux client hosts.
Generate server cache	Specifies you want to generate a shared server cache. If you select this option, TEM runs the generate_client_meta_cache utility at the end of the install, upgrade, or update action. This option reduces Teamcenter memory consumption by moving metadata to shared memory. Types, property descriptors, and constants are placed in a shared cache that is shared by all Teamcenter server instances.
Generate client cache	Specifies you want to generate a cache of data that rich clients can download once at initial logon and then reuse on the client host. This option reduces server demand, reduces startup time, and improves overall performance. When this option is selected, TEM runs the generate_client_meta_cache utility at the end of the install, upgrade, or update action. If you clear this option, but a client cache already exists, the old client cache is deleted.
Production Environment	Specifies your new environment is to be used as a live environment where you will store your product data.
Test Environment	Specifies your new environment is to be used for development, testing, or training. Selecting **Test Environment** enables the bulk loader tool to

Value	Description
	copy data from another environment (such as a production environment) into this test environment. If you designate this environment as a test environment, the designation cannot be changed. Additionally, a test environment cannot participate in Multi-Site sharing with a production environment.

For advanced Teamcenter Foundation options, click **Advanced**.

13. In the **Archive: Teamcenter Administrative User** panel, enter the logon information for the Teamcenter administrative user account.

14. In the **Archive: Flex License Client** panel, enter the settings for the Siemens PLM license server to use for the archive database. By default, the settings used at the production site are specified.

15. The archive site requires Object Directory Services (ODS) and Integrated Distributed Services Manager (IDSM) services be installed and configured. After TEM scans the production and archive sites (or configurations), the **Archive: Multi-Site Configuration** panel displays details on ODS and IDSM services found installed on the sites (if any).

    If ODS or IDSM services are not installed on the archive site, check **Object Directory Services (ODS)** and **Distributed Services Manager (IDSM)** to have them installed.

16. In the **Archive: Multi-Site: Object Directory Services** panel, verify or update the ODS-related details.

17. In the **Archive: Multi-Site: Distributed Services Manager** panel, verify or update the IDSM-related details.

18. When the archive database already exists, TEM will verify the production database and archive database are synchronized. In the **Product/Archive Sync Check** panel, click **Next**. TEM reports any differences between the databases.

19. In the **Confirmation** panel, review the configuration information. Click **Back** to update any setting choices or click **Start** to begin the installation and configuring the archive database.

20. Once the TEM installation completed, perform the following steps to enable registration of archived data to prevent duplication of archived data.

    a. Enable the central item registry as described in Enabling the central item registry.

    b. Set the **TC_4gd_registry_site** preference to the ODS registry site name used for registering 4GD objects.

c. Enable the 4GD item registry by setting the following preference values to **true**:

- **TC_enable_4gd_archive_restore_operation**
- **TC_4gd_registry**
- **TC_4gd_always_register_on_creation**
- **TC_4gd_unregister_on_delete**
- **TC_4gd_allow_creation_if_registry_down**

In addition to the manually-set reference preferences, TEM sets the following preferences:

TC_ods_site_nodes	**TC_idsm_site3_prog_number**
TC_idsm_site_nodes	**TC_ods_site1_prog_number**
ODS_site	**TC_ods_site2_prog_number**
ODS_searchable_sites	**TC_ods_site3_prog_number**
ODS_permitted_sites	**ITEM_id_registry**
IDSM_permitted_sites	**ITEM_id_registry_site**
IDSM_permitted_transfer_sites	**ITEM_id_always_register_on_creation**
IDSM_permitted_checkout_sites	**ITEM_id_unregister_on_delete**
TC_publishable_classes	**TC_STAGING_AREA_SIZE**
TC_transfer_area	**TC_STAGING_AREA**
TC_idsm_site1_prog_number	**TC_4gd_registry_site**
TC_idsm_site2_prog_number	**TC_AR_Excluded_Objs_Folder**

Improving network performance when archiving data

Once archiving is enabled and in use, you may find that transferring the large amounts of data involved in archiving impacts Multi-Site performance. If you incur network timeouts, consider adjusting the timeout values of the following preferences:

- **TC_ods_client_def_timeout**
- **TC_ods_client_initial_timeout**
- **TC_idsm_client_initial_timeout**
- **TC_idsm_client_def_timeout**

- **IDSM_ft_server_timeout**
- **IDSM_ft_client_timeout**

Convert an existing site into an archive site

If you have an existing Multi-Site server you wish to designate as an archive server, be aware of the following items:

- Ensure that no critical data or published records exist on the site prior to designating it as archive site.
- Designating a site as an archive site restricts users from exporting or importing to and from that site.
- Users cannot perform publish and unpublish operations on the objects belonging to archive site.

Use the following process to designate that site as an archive site for one or more production sites in the Multi-Site federation.

1. Identify a site in the federation that will be designated as the archive site. For example, in a federation of sites A, B, and C, site C will become the archive site.
2. Log on to one of the production sites in the Multi-Site federation as a user with administrative and Access Manager bypass privileges.
3. On the archive site and each production site that will use the archive site (log on in remotely if necessary), run **site_util** identifying site C as the archive site. For example:

```
site_util -f=modify -site_id-<site_D_id> -archive=y
```

Transitioning to using TC XML-based Multi-Site Collaboration

TC XML-based Multi-Site Collaboration transition overview

Multi-Site Collaboration leverages TC XML as the underlying data exchange mechanism for all data models. This approach provides significant performance improvements over previous versions of Multi-Site Collaboration which exchanged non-4GD data in binary payloads. These previous versions of Multi-Site Collaboration are no longer supported.

Upgrading to TC XML

New installations of Multi-Site Collaboration use TC XML-based Multi-Site Collaboration by default. However, sites upgrading from versions of Multi-Site Collaboration delivered with Teamcenter 12.0 and earlier (referred to as "legacy Multi-Site Collaboration") must transition to using TC XML-based Multi-Site Collaboration. Transitioning requires **enabling TC XML-based Multi-Site Collaboration and deploying certain configuration changes**. You can transition sites to TC XML-based Multi-Site Collaboration before or after upgrading Teamcenter.

Review **the differences between legacy and TC XML-based Multi-Site Collaboration** and consider the impacts to your organization before transitioning to TC XML-based Multi-Site Collaboration. Thoroughly plan your transition, update command and utility uses, and review and update closure rules, property sets, option sets, and transfer modes.

Multi-Site Collaboration platform consistency requirements

All sites in a Multi-Site Collaboration network must use the same Multi-Site Collaboration technology. That is, all sites must use legacy Multi-Site Collaboration or all sites must use TC XML-based Multi-Site Collaboration. The technologies cannot be used simultaneously in the same TC XML-based Multi-Site Collaboration network.

Sites originally using legacy Multi-Site Collaboration that have transitioned to using TC XML-based Multi-Site Collaboration can switch back to using legacy Multi-Site Collaboration as described in **Manually enable TC XML-based Multi-Site Collaboration**. Siemens Digital Industries Software recommends you do not switch a site back to using legacy Multi-Site Collaboration once it has been transitioned to using TC XML-based Multi-Site Collaboration.

Enable TC XML-based Multi-Site Collaboration

You can use the **multisite_transition_util.exe** utility to transition your sites to use TC XML-based Multi-Site Collaboration. With the utility, you can perform the following tasks:

- Generate and review reports detailing the changes that will occur during an actual transition.
- Migrate your local site to use TC XML-based Multi-Site Collaboration. (Optionally, you can use the information in the earlier-generated reports as guidance to manually transition your sites.)
- Migrate remote sites to use TC XML-based Multi-Site Collaboration. For offline sites, the utility can generate configuration files to be delivered to the remote sites to transition them to use TC XML-based Multi-Site Collaboration.

Alternatively, **you can manually transition your site to use TC XML-based Multi-Site Collaboration**.

Generate and review reports

Before performing a migration, run the **multisite_transition_util.exe** utility with the **f=dryrun** option to generate reports of the changes the utility would make when migrating. You can review these reports to understand the impact on your site when transitioning to TC XML-based Multi-Site Collaboration.

1. Open a command shell and navigate to: *TC_ROOT\bin*
2. Enter the following command:

```
multisite_transition_util -u=user -p=pword -f=dryrun
```

Arguments:

user

A user on the site with administration privileges.

pword

The specified user's password.

3. Answer "**Y**" to each of the utility's prompts. The utility runs. When the utility completes its run, it displays the name of the directory containing the following generated reports:

preference_changes_report.txt

Identifies the preference changes that would occur in an actual migration.

prefs_to_closure_rule_generation_report.txt

Details the relations that would be converted to clauses. (The clauses may require adjustments after running the utility, but those adjustments may be minimal compared to manually creating the clauses.)

4. Review the reports to understand the adjustments necessary when transitioning to TC XML-based Multi-Site Collaboration, particularly the changes necessary to convert relation types to clauses. If the adjustments are minimal, using the utility is a better option than manually recreating all the clauses.

Migrate the local site

Run the **multisite_transition_util.exe** utility with the **f=migrate** option to transition your local site.

1. Before migrating the site, back up your current Multi-Site closure rule settings using the following command:

```
tcxml_export -u=user -p=pword -uid=Multi-SiteCRUID -file=backupfile
```

Arguments:

user

A user on the site with administration privileges.

pword

The specified user's password.

uid

The unique ID of your **MultiSiteDefaultCR** closure rule file.

backupfile

The name of the backed up closure rule file.

2. From a command shell at the local site, navigate to: *TC_ROOT\bin*

3. In the command shell, enter the following command:

   ```
   multisite_transition_util -u=user -p=pword -f=migrate
   ```

   Arguments:

   user

   A user on the site with administration privileges.

   pword

   The specified user's password.

4. The utility prompts you to answer several transition-related questions and then runs. When the utility completes its run, it displays the name of the directory containing reports of the changes made when transitioning the site.

5. Review the following sections and adjust your site's configuration as necessary.

   - **Comparison of legacy and TC XML-based Multi-Site Collaboration**
   - **Import and export relation types**
   - **Transfer data between sites that use different schemas**

Migrate remote sites

Use the following processes to transition remote sites on your Multi-Site Collaboration network to use TC XML-based Multi-Site Collaboration.

Transition online sites

Run the **multisite_transition_util.exe** utility with the **f=push** option to transition an online remote site in your Multi-Site Collaboration network. The utility updates the remote site with the same changes as made to the local site.

1. Before migrating the remote site, back up the current Multi-Site closure rule settings at the remote site using the following command:

   ```
   tcxml_export -u=user -p=pword -uid=Multi-SiteCRUID -file=backupfile
   ```

   Arguments:

user

A user on the site with administration privileges.

pword

The specified user's password.

uid

The unique ID of your **MultiSiteDefaultCR** closure rule file.

backupfile

The name of the backed up closure rule file.

2. At the local site (already transitioned to using TC XML-based Multi-Site Collaboration), open a command shell and navigate to: *TC_ROOT\bin*

3. In the command shell, enter the following command:

   ```
   multisite_transition_util -u=user -p=pword -f=push -site=target_site
   ```

   Arguments:

   user

   A user on the local site with administration privileges.

   pword

   The specified user's password.

   target_site

   The remote site to be transitioned.

4. The utility prompts you to answer several transition-related questions and then runs. When the utility completes its run, it displays the status of the remote site's transition.

5. Review the following sections and adjust the remote site's configuration in the same manner as done for the local site.

   - **Comparison of legacy and TC XML-based Multi-Site Collaboration**
   - **Import and export relation types**
   - **Transfer data between sites that use different schemas**

Transition offline sites

Use the following process for transitioning offline remote sites in your Multi-Site Collaboration network. The process updates the remote site with the same changes as made to the local site.

1. Before migrating the remote site, back up the current Multi-Site closure rule settings at the remote site using the following command:

   ```
   tcxml_export -u=user -p=pword -uid=Multi-SiteCRUID -file=backupfile
   ```

   Arguments:

   user

   A user on the site with administration privileges.

   pword

   The specified user's password.

   uid

   The unique ID of your **MultiSiteDefaultCR** closure rule file.

   backupfile

   The name of the backed up closure rule file.

2. At the local site (already transitioned to using TC XML-based Multi-Site Collaboration), open a command shell and navigate to: *TC_ROOT\bin*

3. In the command shell, enter the following command:

   ```
   multisite_transition_util -u=user -p=pword -f=export
   ```

   Arguments:

   user

   A user on the site with administration privileges.

   pword

   The specified user's password.

4. The utility prompts you to answer several transition-related questions and then runs. The utility runs, exporting the preference and closure changes necessary to transition a remote site.

   Note the location of the export folder.

5. Copy the export folder and its contents to the remote site.

6. Back up the current Multi-Site closure rule settings at the remote site using the following command:

   ```
   tcxml_export -u=user -p=pword -uid=Multi-SiteCRUID -file=backupfile
   ```

Arguments:

user

A user on the site with administration privileges.

pword

The specified user's password.

uid

The unique ID of your **MultiSiteDefaultCR** closure rule file.

backupfile

The name of the backed up closure rule file.

7. In a command shell at the remote site, navigate to: *TC_ROOT\bin*

8. In the command shell, enter the following command:

```
multisite_transition_util -u=user -p=pword -f=import
```

Arguments:

user

A user on the site with administration privileges.

pword

The specified user's password.

9. Answer "**Y**" to the prompts to import preferences and closure rules and provide the paths and file names of the exported preference and closure rule files.

10. Review the following sections and adjust the remote site's configuration in the same manner as done for the local site.

- **Comparison of legacy and TC XML-based Multi-Site Collaboration**
- **Import and export relation types**
- **Transfer data between sites that use different schemas**

Manually enable TC XML-based Multi-Site Collaboration

Siemens Digital Industries Software recommends you **use the multisite_transition_util.exe utility to transition your site to use TC XML-based Multi-Site Collaboration**. If using that utility is not

appropriate for your site, use the following guidelines to manually transition your site to use TC XML-based Multi-Site Collaboration

1. Delete the site preference **TC_force_legacy_multisite**. (This preference is deprecated and will not be supported in a future release.)

2. Ensure the site preference **IDSM_Compression** is not set to **true**. **IDSM_Compression** is not supported by TC XML-based Multi-Site Collaboration.

3. Optionally retain legacy Multi-Site Collaboration replication and propagation behavior by setting the **TC_legacy_multisite_propagation** preference to a value of **true**. See **Replication and propagation differences** for more information.

4. Review the following sections and adjust your site's configuration as necessary.

   - **Comparison of legacy and TC XML-based Multi-Site Collaboration**
   - **Import and export relation types**
   - **Transfer data between sites that use different schemas**

Siemens Digital Industries Software recommends once sites are using TC XML-based Multi-Site Collaboration, they do not revert back to using legacy Multi-Site Collaboration. However, if sites must revert to using legacy Multi-Site Collaboration, they can do so by setting the following preference values:

- Set **TC_force_legacy_multisite** to **true**.
- Set **TC_island_cico_scope** to no value.

Comparison of legacy and TC XML-based Multi-Site Collaboration

Command line utility differences

Though most TC XML-based Multi-Site Collaboration utilities operate the same as with legacy Multi-Site Collaboration, certain utilities are not supported with TC XML-based Multi-Site Collaboration. Replace uses of those utilities as follows.

Legacy Multi-Site Collaboration	TC XML-based Multi-Site Collaboration
data_share	Same use as legacy Multi-Site Collaboration.
data_sync	Same use as legacy Multi-Site Collaboration.
sync_on_demand	Same use as legacy Multi-Site Collaboration.
Item_export	Replace uses with **data_share -f=offline_export**.
Item_import	Replace uses with **data_share -f=offline_import**.
ensure_site_consistency	Same use as legacy Multi-Site Collaboration.

Legacy Multi-Site Collaboration	TC XML-based Multi-Site Collaboration
export_recovery	Replace uses with **ensure_site_consistency -f=offline_recovery**.
migrate_organization	Same use as legacy Multi-Site Collaboration.

Rich client import and export options

When using TC XML-based Multi-Site Collaboration, import and export options are supported by default with the following closure rules, property sets, option sets, and transfer modes to scope the transferred data.

- **MultiSiteExpOptSet** (optionset)
- **MultiSiteDefaultCR** (closure rule)
- **MultiSiteAdmin_PS** (property set)
- **MultiSiteDefaultTM** (transfer mode)

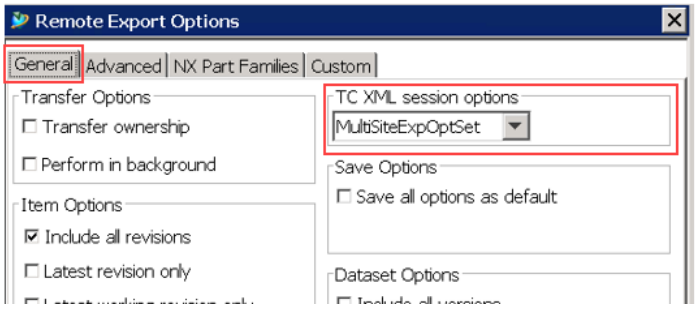
Modify these or create your own versions to support your needs. When creating your own versions, set **-optionset** to your own option set instead of using the default option set, **MultiSiteExpOptSet**. When creating a new option set, add the **opt_multi_site** option to the option set and set it to **true**.

Use the same option set, closure rules, property set, and transfer mode with all sites in the Multi-Site federation. Use PLM XML/TCXML Export Import Administration to manage this data. See *Tc XML and PLM XML Configuration for Data Import and Export*.

Rich client dialog box differences

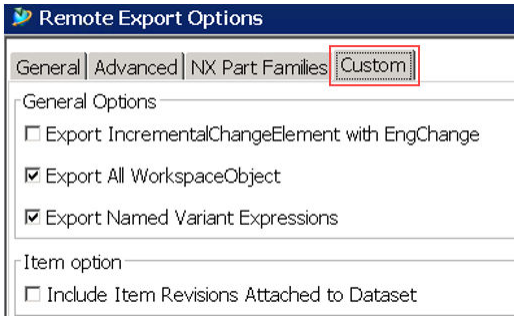
When using TC XML-based Multi-Site Collaboration, the **Remote Export Options** dialog box differs from the same dialog box when using legacy Multi-Site Collaboration as follows.

- On the **General** tab, the **TC XML session options** list lets you pick the option set to use.



- On the **Advanced** tab, the **Relationship Options** are disabled. Set up closure rules to specify references to include and exclude.

- A **Custom** tab is available, providing configuration options based on the option set selected in the **TC XML session options** on the **General** tab as in the following example.



Using the ensure_site_consistency utility

When performing corrective recovery actions using the ensure_site_consistency utility, be aware of the following differences in usage between legacy Multi-Site Collaboration and TC XML-based Multi-Site Collaboration.

To restore ownership on an item with the ID MyCorruptItem:

Using legacy Multi-Site Collaboration:

```
export_recovery -mode=auto -item_id=MyCorruptItem
-remote_site=Manufacturing
```

Using TC XML-based Multi-Site Collaboration:

```
ensure_site_consistency -mode=auto -f=offline_recovery
-item_id=MyCorruptItem
-remote_site=Manufacturing
```

To restore ownership on objects contained in an export metafile without reimporting:

Using legacy Multi-Site Collaboration:

```
export_recovery -mode=min -dir=metafile_dir
```

Using TC XML-based Multi-Site Collaboration:

```
ensure_site_consistency -mode=min -f=offline_recovery
-dir=tcxmlfile_dir
```

To reimport objects from the metafile and restore site ownership:

Using legacy Multi-Site Collaboration:

```
export_recovery -mode=full -dir=metafile_dir
```

Using TC XML-based Multi-Site Collaboration:

```
ensure_site_consistency -mode=full -f=offline_recovery
-dir=tcxmlfile_dir
```

To make an item (for example, xyz) in the local site a replica that is owned by another site (for example, Site2):

Using legacy Multi-Site Collaboration:

```
export_recovery -mode=auto -item_id=xyz -real_owning_site=Site2
```

Using TC XML-based Multi-Site Collaboration:

```
ensure_site_consistency -mode=auto --f=offline_recovery item_id=xyz
-real_owning_site=site2
```

To restore ownership of an entire assembly:

Using legacy Multi-Site Collaboration:

```
export_recovery -mode=auto -item_id=Assy1 -remote_site=Site2
-include_bom
```

Using TC XML-based Multi-Site Collaboration:

```
Ensure_site_consistency -mode=auto --f=offline_recovery item_id=Assy1
-real_owning_site=site2 -include_bom
```

Import and export relation types

By default, only required relation types are exported. To export additional relation types, create additional closure rules for the types.

For example, the following closure rule collects all relation types when the **opt_exp_all_wso** option is set to **true** in the option set.

```
WorkspaceObject:WorkspaceObject:P2S:PROCESS+TRAVERSE:$opt_exp_all_wso!
="false":I
```

Be aware that a closure rule such as the one in the example will collect all relation types, potentially significantly decreasing system performance.

Following is a comparison of include, exclude and relation types between legacy Multi-Site Collaboration and TC XML-based Multi-Site Collaboration.

Legacy Multi-Site Collaboration	TC XML-based Multi-Site Collaboration
Command line support of **–include** and **-exclude** parameters.	Supported through closure rule setup.
Select and deselect relation types.	Supported through closure rule setup.
Relation types are exported using the following preferences **TC_relation_required_on_export** **TC_relation_required_on_transfer**	By default, relations delivered with Multi-Site Collaboration specified by these preferences have been converted to closure rules. Manually create closure rules for any custom relations created at your site. Relation preferences support is deprecated. Sites must set up external closure rules to cover additional relations.

Use the following process and examples as a guide when setting up include and exclude relation closure rules. The example scenario creates a closure rule for a site that previously had **IMAN_Rendering** specified by the **TC_relation_required_on_export** preference.

1. Set up the **IMAN_Rendering** export conversion rules. For example:

   To always include IMAN_Rendering:

   Add the following external closure rule:

   ```
   ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse
   ```

   If setup opt_excl_rendering is set to false, include IMAN_Rendering:

   a. Add the following external closure rule:

      ```
      ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse:
      opt_excl_rendering=="false"
      ```

   b. Add the **opt_excl_rendering="false"** option to **MultiSiteExpOptSet**.

   c. To optionally exclude the relation when running **data_share**, provide the parameter **-override_options=opt_excl_rendering:true**.

   If setup opt_excl_rendering is set to true, exclude IMAN_Rendering:

   a. Add the following external closure rule:

      ```
      ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse:
      opt_incl_rendering=="true"
      ```

   b. Add the **opt_incl_rendering="false"** option to **MultiSiteExpOptSet**.

   c. To optionally include the relation when running **data_share**, provide the parameter **-override_options=opt_incl_rendering:true**.

2. Set **TC_relation_required_on_transfer** to a value of **IMAN_Rendering**.

3. Set up the **IMAN_Rendering** transfer conversion rules. For example:

   To exclude the relation on replication, but only export on ownership transfer:

   Add the following external closure rule:

   ```
   ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse:
   opt_xfer_ownership=="true"
   ```

   To always include IMAN_rendering on both ownership transfer and replication:

   Add the following external closure rule:

   ```
   ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse:
   ```

   To always exclude IMAN_Rendering, unless transferring ownership, or when opt_incl_rendering is set to true:

   a. Add the following external closure rule:

      ```
      ItemRevision:Dataset:P2S:IMAN_Rendering:Process+Traverse:
      opt_incl_rendering=="true" || opt_xfer_ownership=="true"
      ```

   b. Add the **opt_incl_rendering="false"** option to **MultiSiteExpOptSet**.

   c. To optionally include the relation when running **data_share**, provide the parameter **-override_options=opt_incl_rendering:true**.

Transfer data between sites that use different schemas

Sites using different schemas may use different objects to represent the same data. For example, you may need to map Site 1's use of Item to Site 2's use of Design. If you are transferring data between sites using different schemas, use the Advanced Multi-Schema Exchanger to create mapping rules to use when transferring data between sites.

Use the following process and examples to as guidance in creating your data mapping.

1. Set up your mapping on Site 2, using the Advanced Multi-Schema Exchanger to map the objects in Site 1's schema to the objects in Site 2's schema.

2. On Site 2, attach the transformation rules to a transfer mode, for example, **Site1ToSite2XferMode**. Attach the transformation rules using a command such as in the following example:

   ```
   plmxml_tm_edit_xsl -transfermode=Site1ToSite2XferMode -action=attach
   -xsl_file=transformerFile
   ```

3. On Site 2, set the **TC_tms_site_interop_transfer_mode** site preference with the values **Site1**, **Site1ToSite2XferMode**.

When exporting data from Site 1, the objects will be converted based on the rules defined in the mapping rules file attached to **Site1ToSite2XferMode**.

Configuring propagation with TC XML-based Multi-Site Collaboration

Siemens Digital Industries Software recommends that propagation rules be the same at the owning and all replication sites. However, if your Multi-Site federation requires using different propagation rules at owning and replication sites, you can still perform propagation and replication. To do so, set the **TC_legacy_multisite_propagation** preference to **true**. When setting this preference to **true**, replication occurs and rules on the remote site that differ from the owning site are followed. However, replication will take longer than when rules are the same at all sites and this preference is not set to **true**.

Be aware that TC XML-based Multi-Site Collaboration does not support **ProjectObjectRelation** objects (PORs) with **Propagation Group** set to "No Group". If you have these PORs on your site, use the **migrate_propagation_data** utility to migrate them before using them with TC XML-based Multi-Site Collaboration.

For more information on working with program and project data with Multi-Site, see Considerations for importing and exporting project or program data in a Multi-Site environment.

Configuring licensing with TC XML-based Multi-Site Collaboration

ADA license for IP

For information on configuring Multi-Site for authorized data access (ADA) control using an intellectual property (IP) license, see Multi-Site Collaboration considerations.

ADA ITAR support

For information on configuring Multi-Site when configuring authorized data access for International Traffic in Arms Regulations (ITAR) support, see Multi-Site Collaboration considerations for ITAR.

Batch processing appearance path nodes (APNs) objects using TC XML-based Multi-Site Collaboration

TC XML-based Multi-Site Collaboration supports batch processing **VariantExpression** and **MEAppearance PathNode** objects using the **data_share** and **data_sync** utilities.

Do not use the **-batch_objects** option of either utility to process **Dataset**, **ImanRelation**, **PSOccurrence**, **Folder**, **Form**, **NamedVariantExpression**, and **VariantExpressionBlock** objects.

Use one of the following strategies to process APNs:

Send all APNs with an item in a single use of data_share

Perform a basic import using the **data_share** utility, during which TC XML-based Multi-Site Collaboration processes added and removed APNs. No APNs remain at the import site. Siemens Digital Industries Software recommends this strategy.

For example:

```
data_share -item_id=xxxx -f=send -site=target_site
```

Send APNs using the data_share -batch_object option

Process the item using the **data_share** utility, sending the APNs using batch mode. The batch processing will not infer that additional APNs found at the importing site no longer exist. Therefore, orphan APNs will remain at the import site. Use a command with the following form:

```
data_share -item_id=xxxx -f=send -site=target_site
-batch_objects=MEAppearancePathNode
```

Send the item using data_share and send APNs using sync_product_apns

1. Skip APN import when running **data_share** by either setting **TC_EXCLUDE_APN** to **true**, or by running **data_share** with **-override_options** set to **opt_exclude_apn:true**. For example:

   ```
   data_share -item_id=xxx -f=send -site=target_site
   -override_options=opt_exclude_apn:true
   ```

2. Use **sync_product_apns** with one of the following methods to send a list of APNs:

   Online process

   Send the list of APNs using a command with the following form:

   ```
   sync_product_apns -send -item=xxxx -site=yyyy
   ```

   Offline export and import

   Perform an offline export using a command with the following form:

   ```
   sync_product_apns -item=xxx -dir=outputDir -site=yyyy
   ```

   Perform an offline import using a command with the following form:

   ```
   sync_product_apns -item=xxx -dir=inputDir
   ```

   sync_product_apns with data_share

   Use **sync_product_apns** to generate an APN UID list. Then, fetch the list using **data_share**.

   Generate the APN UID list using a command with the following form:

   ```
   sync_product_apns -item=xxx -count -tag_list=APNUidListFile
   -site=target_site
   ```

   The APN UID list has the following form:

```
3 <- count
APNUid001
APNUid002
APNUid003
```

Before fetching this list, remove "`<- count`" from the file. (Leaving it in the file results in a **data_share** failure.) Fetch the list using a command with the following form:

```
data_share -filename=APNUidListFile -f=send -site=target_site
-classoffile=Tagstring
```

Process for deleting unwanted replicas

Replica deletion process

The replica deletion process removes unnecessary replicated data in a Multi-Site federation and reduces the associated infrastructure cost. The replica deletion process requires extensive planning and uses a variety of Siemens Digital Industries Software tools.

> Warning:
>
> Some of the tools and actions described in this process can cause data loss if they are not used or performed properly. Because of potential damage to your enterprise data, you must ensure that you have the proper experience to perform the required tasks. You must fully understand the intent of each process step and the use of each tool before attempting to reduce replicated data in a Multi-Site federation.

> Caution:
>
> Replica deletion procedures are complex and require precision planning and execution.
>
> Following these processes and using the **delete_replica** utility do not guarantee success. Data corruption can occur prior to, during, or after deletion. A well-planned methodology developed by an experienced system integrator is essential.

All involved sites (source, target sites) must be operating at the same appropriately licensed Teamcenter version level and must include the **delete_replica** utility. The involved target sites are those that have shared Teamcenter data with the source site. The **delete_replica** utility must be run at each target site during the execution phase.

A typical large enterprise may have multiple sites in a Multi-Site federation. The replica deletion process is organized in three distinctive phases:

Preparation During the *preparation* phase, you identify data, do in-depth analysis, and perform specific readiness steps to ensure that product and volume data are ready for deletion. This task must be carried out at all sites in the Multi-Site federation environment.

All involved sites (source, target sites) must be operating at the same appropriately licensed Teamcenter version. This version must include the **delete_replica** utility. The involved target sites are those that have shared Teamcenter data with the source site. You can determine the sites using the MSA sharing profile analysis function. The **delete_replica** utility must be run at target sites during the execution phase.

Execution During the *execution* phase, you delete the replicated product data from the target site. You delete the identified replicated product data, in one or more increments, from the target site using multiple batches depending on data size or volume of objects.

A best practice is to perform a dry run using **delete_replica** utility to generate a report prior to performing that actual replica deletion. You can user the bar (**|**) character as a delimiter to import the report into Microsoft Excel. Review the report for displayed errors and validate that the execution of the utility deletes the appropriate identified replicated data and its dependent objects.

Verification During the *verification* phase, you review the execution phase report and system logs (**syslog**) files for any errors to determine if any corrections are required.

Prepare Multi-Site sites for deleting unneeded replicas

The preparation phase requires extensive data mining and analysis. Depending on the size and complexity of the effort, it may take weeks to develop and refine a definitive plan.

1. Deploy the additional components and the information technology (IT) environment required by the replica deletion activity. The replica deletion process uses the default **SiteConsolidationDefault** and **SiteConsolidationLW** transfer option sets.

    a. Launch the Teamcenter rich client and log on as a user with administrator privileges. Open the PLM XML/TC XML Export Import Administration application.

    b. Ensure the **SiteConsolidationDefault** transfer option set is associated with **SiteConsolidationDefaultTM** transfer mode consistently at the source and target sites.

    c. Ensure the **SiteConsolidationDefaultTM** transfer mode output schema format is set to **TC XML** and it is associated with the **SiteConsolidationDefaultCR** closure rule consistently at the source and target sites.

    d. Ensure the **SiteConsolidationDefaultCR** closure rule output schema format is set to **TC XML**.

    e. Ensure the **SiteConsolidationLW** transfer option set is associated with **SiteConsolidationLWTM** transfer mode consistently at the source and target sites.

    f. Ensure the **SiteConsolidationLWTM** transfer mode output schema format is set to **TC XML** and it is associated with the **SiteConsolidationLWCR** closure rule consistently at the source and target sites.

    g. Ensure the **SiteConsolidationLWCR** closure rule output schema format is set to **TC XML**.

If there are local customizations to the data model, you may have to extend and/or modify the closure rules to ensure the instantiated custom data is extracted.

2. Using out-of-the-box (OOTB) search queries or custom queries, identify the top-level object of the replicated product data to delete. You can use various data attributes and filters, such as project type, owning user, owning group, release status, creation date, last modified date, and so forth, to identify the object.

   Use the Query Builder application to create the required custom queries.

3. Find any data issues in the identified data at both the source and target site. Determine if any of the issues impact the correct functioning of the **delete_replica** utility when it is used to delete the identified data. You can use the following tools:

   plm_report_extract utility

   Use to extract the persistent data in binary format from a Teamcenter site. This extract can be used to perform consistency analysis with similar extracts from other Teamcenter sites using the **plm_report_consistency_analysis** utility.

   plm_report_consistency_analysis utility

   Use primarily to compare the results of multiple site extracts, analyze that comparison, identify Multi-Site related issues, and suggest fixes for the issues found. It is based on the extracts from multiple sites. It also provides the ability to convert the proprietary extract file (binary file) into a text file that can be used by other applications, such as Microsoft Excel.

4. Correct any issues using standard Teamcenter utilities, such as the **export_recovery**, **item_rename**, and **item_relink** utilities.

Delete unneeded replicas

1. Generate a report of the objects to be deleted using the list function of the **delete_replica** utility, for example:

   ```
   delete_replica -u=cv1admin -p=tyad17FY f=list -verbose
   -itemidsfile=item_list.txt -batch-5 -delete_exportrecords
   ```

   The **delete_replica** utility accepts input replica objects of type item, item revision, dataset, and form.

   The **-itemidsfile** argument requires to full path and file name of a file containing a list of replica item IDs. They may be listed one per line or on one line separated by commas.

2. Review the report for any errors or anomalies. You can import the file into an Microsoft Excel spreadsheet for easier viewing using a pipe or bar (**|**) character as the delimiter.

3. Validate that the correct replicated data and its dependent objects and only these object are listed in the report for deletion.

4. Delete replicas using the delete function of the **delete_replica** utility, for example:

```
delete_replica u=cv1admin -p=tyad17FY -f=delete -verbose
-itemidsfile=item_list.txt -batch=35 -delete_exportrecords
-delete_volume_files -include_bom
-report_file=deletion_report.txt
```

   When deleting replicas in multiple batch increments, stub objects may be created to maintain referential integrity of objects across islands of data. Once all the islands of data are deleted, several of the stubs become unreferenced and are consequently deleted at the end of replica deletion process.

Using the delete_replica utility

The **delete_replica** utility accepts input replica objects of type item, item revision, dataset, form, and 4GD. The input objects are validated for various error conditions and traversed using site consolidation closure rules to determine the dependent objects. The dependent objects are subjected to internal rules to provide a list of objects that can be deleted. You can perform a dry run and generate a report without affecting the database. When supplied, 4GD arguments take precedence in processing over other arguments.

The utility allows specifying the input options as item ID, item key, item ID list, object UID list, and batch size. By default, the utility processes a batch size of 25 input objects (islands) in a single pass. However, you can set a higher or lower batch size based on available memory resources.

The input objects are validated for error conditions as follows:

- You have user administrative privileges.
- Input objects exists and are replicas.
- Input objects are of type **Item**, **ItemRevision**, **Dataset**, or **Form**.
- Input objects are not of type item primary (master) form or item revision primary form.

Any input object that does not meet the validation criteria is reported and excluded from further processing. The remaining set of input objects are traversed using site consolidation closure rules to determine dependent objects.

The dependent objects are validated as follows:

- Validate dataset references.

Operations, such as save as and revise, may provide datasets as references. If the dataset has item or item revision references across islands of data and is not part of the current batch of objects, the dataset object is excluded from further processing.

- Validate revision anchor references.

  If the revision anchor has references across islands of data and is not part of the current batch of objects, the revision anchor object is excluded from further processing.

- Validate release status references.

  If a release status has references across islands of data and is not part of the current batch of objects, the release status object is excluded from further processing.

- Verify whether the dependent objects is part of a workflow.

  If yes, a validation error is reported for the object in workflow and the entire island of data is excluded from further processing.

- Verify whether the dependent objects are checked out.

  If yes, a validation error is reported for the checked-out object and the entire island of data is excluded from further processing.

- Analyze the dependent objects for mixed (local and remote) ownership.

  Local objects of the following types are included for further processing and deletion:

  - **IMAN_RES_audit**
  - **IMAN_RES_checkout**
  - **IMAN_based_on**
  - Local lightweight objects (LWOs) belonging to the class and subclass of **Fnd0AuditLog**

  ImanFile objects with physical volume file names starting with **audit** and **cico** are included for further processing and deletion.

  Local objects related to the following types are excluded for further processing:

  - **ProjectObjectRelation**
  - Effectivity (shared effectivity)

If any other type of local object is referenced by the replica, the entire island of data is excluded from further processing.

- Evaluate the dependent workspace objects for any folder references and if found, delete the folder references.

The utility provides additional options to:

- Configure the option set.
- Delete volume files.
- Delete export records.
- Delete child components.
- Exclude standard and catalog items from deletion.
- Set report file location.

You can provide your own closure rules and associate them with a transfer option set (TOS). You specify the option set with the **-optionset** argument of the **delete_replica** utility. The default TOS is the **DeleteReplicaOptionSet** TOS (derived from the **SiteConsolidationDefault** TOS).

The delete export records option removes IMAN export records and item export records associated with replicas at owning site.

If any local components are found in the subassembly, they are not deleted. However, the local components are removed from the subassembly product structure before the subassembly is deleted.

delete_replica utility syntax

```
delete_replica -u=<user_id>{-p=<password> | -pf=<password file>}
-g=<group>
-f={ list | delete }
[-item_id=<item_id>]
[-rev=<rev_id>]
[-object_uid=<uid>]
[-key=<item_key>]
[-itemidsfile=<data_file>]
[-itemKeyKile=<data_file>]
[-itemRevisionKeyFile=<data_file>]
[-uidsfile=<data_file>]
[-optionset=<name>]
[-delete_volume_files]
[-delete_exportrecords]
[-include_bom]
```

```
[-batch=<number>]
[-report_file=<report_file_name>]
[-exclude_ids=<file_name>]
[-4gd_id]
[-class=<wso_class_name>]
[-classoffile=<class_name>]
[-verbose]
[-h]
```

For a description of each argument, see the utility's help (**-h**) output.

delete_replica utility examples

Required logon information is omitted from the following examples.

- To generate a dry run report of dependent objects to be deleted for a given replica item ID:

  ```
  delete_replica -f=list  -verbose -item_id=ABC_101
  ```

- To generate dry run report for a given input file of item ID strings:

  ```
  delete_replica -f=list -itemidsfile=item_list.txt -batch=5 -delete_exportrecords
  -verbose
  ```

- To perform deletion and generate report for a given replica item ID:

  ```
  delete_replica -f=delete -delete_exportrecords -verbose -item_id=ABC_101
  ```

- To perform deletion and generate report for a given input file of item ID strings including assembly components:

  ```
  delete_replica -f=delete -itemidsfile=item_list.txt -batch=35 -delete_exportrecords
  -verbose
  -delete_volume_files  -include_bom -report_file=deletion_report.txt
  ```

- To generate a dry run report for a given 4GD object:

  ```
  delete_replica -f=list -4gd_id=MS1_DE000022 -class="Cpd0DesignElement"
  ```

Multi-Site related workflow handlers

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

- **PUBR-publish-target-objects**
- **OBJIO-send-target-objects**

- **PUBR-unpublish-target-objects**

> Note:
>
> Do not set the **PUBR-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**.

Multi-Site related utilities

The following data sharing utilities support Multi-Site Collaboration functionality. You can use them to aid in eliminating duplicate items across sites. You must be a system administrator user to run these utilities.

> Note:
>
> If you are using the IPEM CAD Manager integration, the **TC_set_ipem_mode_for_remote_import_and_export** preference must be set to **TRUE** to export the required objects.

- **database_verify**
- **export_recovery**
- **data_share**
- **item_export**
- **item_relink**
- **item_rename**

System administration data

Using the dsa_util utility

Use the **dsa_util** utility to distribute system classes.

> Caution:
>
> Do not use the **dsa_util** utility to distribute organization data in a global organization environment.

Use only the **admin_data_export** and **admin data_import** utilities to transfer administration data between sites.

For consistency with other Teamcenter applications, the syntax of the **dsa_util** utility command line switches 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.

When you distribute user and groups, it is important to understand the following aspects of the **dsa_util** 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 **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.

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.

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 email 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 is 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 Digital Industries 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.

Security controls

Site-level security of distributing system administration data is enforced using the **IDSM_dsa_sites_permitted_to_push_admin_data** preference. Each local site must define this preference in its 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.

Migrating organization objects

Migration process and limitations

You use the **migrate_organization** utility to identify duplicate organization data between two sites. After identifying cloned organization objects (objects that exist at both sites and are identical), you use the utility to make the identified objects replicas.

> Note:
>
> By design, the database administrator (DBA) group and DBA role cannot be exported through Multi-Site Collaboration or migrated using the **migrate_organization** utility. Users can be assigned to project teams in conjunction with existing Teamcenter organizational roles. In a global organization environment, you cannot export or migrate the project team assignments.

The migration must be performed using a bottom-up process by migrating the objects in the following order:

1. **Person** objects
2. **User** objects
3. **Role** objects (migrates the associated **GroupMember** objects)
4. **Group** objects

Only one site can be migrated at a time.

> Caution:
>
> Do not run the **migrate_organization** utility from the intended replica site. This utility must be run from the owning site only.

Premigration tasks

> Caution:
>
> Set the **IDSM_global_dsa_sites_permitted_to_push_admin_data** preference before starting the migration. The effort required to roll back undesired changes is significant if this preference is not set correctly prior to using the **migrate_organization** utility.

1. Choose the site to be the owning site for your organization.
2. Select the groups you want migrated to the global organization. Every **Role**, **User**, **Person**, and **GroupMember** class object in the selected groups must be migrated.

3. If a role is shared, migrate all the groups the role exists in.

4. Run the **migrate_organization** utility with the **-f=compare_organization** function and the **-report** argument at the owning site against a target site (site where cloned data becomes replicas) to create a report of duplicate and missing organizational differences.

5. Examine the contents of the created file and identify required organizational objects missing at the owning site and organizational objects that are identical at both sites for the selected organization:

   a. Create any **User** and **Person** objects that exist at the target site that are missing at the owning site.

   b. Make group hierarchies identical in structure and names.

   c. Create any roles in groups that exist at the target site that are missing at the owning site.

   d. Create any group memberships that exist at the target site that are missing at the owning site.

   > Note:
   >
   > **User**, **Person**, and **GroupMember** objects that exist only at the owning site require no action.

6. Run the **migrate_organization** utility with the **-f=compare_organization** function and the **-report** argument again at the owning site to ensure the organization objects are identical at both sites. Ensure that you use a unique report file name for each remote site and save the file for use in generating an input list for the **make_replica** function.

7. Perform steps 5 and 6 for every target site in your organization.

8. Create an input file for each class of object to be replicated at each remote site:

   a. Import the report file from step 6 into Microsoft Excel using the pipe (**|**) character as a custom delimiter.

   b. Select all cells.

   c. Select the **Filter** option and filter in column **A** using the **Text filters→Contains** menu command and type the appropriate class name, that is, **Person**, **User**, **Role**, or **Group**.

   d. Ensure the filtered content is selected, copy and paste the filtered objects into an ASCII editor and save the document as plain text.

Migrate organization data

The **migrate_organization** utility allows one **-site** argument, so each target site must be converted separately. Each **Group**, **Role**, **User**, and **Person** object name to be converted must be specified in the appropriate argument when you run the utility.

Alternatively, you can use **-classofffile** and **-filename** arguments for each of the following steps. However, you must perform the object migration in the order shown.

1. Run the utility specifying the **-person_name** argument with the name of all **Person** objects to be migrated.

   Person objects must be migrated before any associated **User** objects.

2. Run the utility specifying the **-user_id** argument with the user IDs of all **User** objects to be migrated.

   Before migrating a role, the associated **GroupMember** object's users must be migrated.

3. Run the utility specifying the **-role_name** arguments with the names of all **Role** objects to be migrated.

4. Run the utility specifying the **-group_name** argument with the names of all **Group** objects to be migrated.

   Only top-level groups can be migrated. Specifying a subgroup causes a utility error. All subgroups of the target group are migrated.

Post migration requirements

- Changes to organization objects must be made at the owning site.
- Create all new organization objects at the owning site and export them to the remote sites.
- Set up a script to (or manually) run the **data_sync** utility on the owning site to periodically synchronize changes to organization data.
- Do not use the **dsa_util** utility to distribute organization objects that are part of a global organization.

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 **OBJIO-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.

There are several 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** 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.

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.

8. Custom configurations

Configuring multiple sites on a server

Set up multiple ODS daemons on a single Linux server

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 "$TC_ROOT/bin/run_tc_ods &"
   su "$TC_ROOT/bin/run_tc_ods_chicago &"
   su "$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 Linux server

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 Tc-admin-user ${TC_ROOT}/bin/run_tc_idsm
   run_tc_idsm
   ```

   Where *Tc-admin-user* is a user with Teamcenter administrative privileges.

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

   ```
   536875586/1 tli rpc/tcp nowait Tc-admin-user
      ${TC_ROOT}/bin/run_tc_idsm_chicago run_tc_idsm 536875587/
   1 tli rpc/tcp nowait Tc-admin-user ${TC_ROOT}/bin/run_tc_idsm_detroit
   run_tc_idsm
   ```

   Where *Tc-admin-user* is a user with Teamcenter administrative privileges.

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 **/etc/init.d** directory on Linux platforms.

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

Set up multiple ODS processes on a single Windows 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 processes are 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 processes are now running on the same service.

Set up multiple IDSM processes on a single Windows 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 that control the IDSM processes are now modified. You must have a utility that can add and delete Windows services. The following example uses the **sc.exe** program. 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_idsm_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 processes are now running on the same service.

Configuring preferences for multiple sites on a single server

Configure the 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.

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** 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.

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

For example, add the following 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

Methods for communicating 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 enabled 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 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 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.

Configuring Multi-Site for RPC communications

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.

Linux:

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

   Make these edits on both the proxy server and the 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.

IDSM launching utility

On Linux 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**.

Configure the IDSM on Linux systems

Use the following steps to configure the IDSM on Linux systems.

1. Create the directory *TC_ROOT*/*multisite* if it does not exist.

2. Create the **tc_idsminetd.service** script in *TC_ROOT*/*multisite* using the following example:

```
<begin script>
[Unit]
Description=Teamcenter idsminetd Service

# Unit starts up after machine connects to network
After=network.target

[Service]
# idsminetd forks and execs
Type=forking

# Command to execute when the service is started
ExecStart=/apps/tc/tc14/TR/bin/idsminetd -p=1086 -r=/apps/tc/
tc14/TR/bin/run_tc_idsm

# Service will always restart when the process exits, is killed, or a
timeout is reached
Restart=always

[Install]
# Necessary for keeping this unit enabled by default
WantedBy=multi-user.target

<end script>
```

3. Modify the `ExecStart` line so the full path to **idsminetd** and **run_tc_idsm** are correct for your installation. (Both are located in *TC_ROOT*/*bin*.)

4. Modify the `ExecStart` line so `-p` is set to your selected IDSM port number.

5. Run the following commands to have the service run at system startup and to immediately start the service:

```
sudo systemctl enable $TC_ROOT/multisite/tc_idsminetd.service
sudo systemctl start tc_idsminetd.service
```

Managing the tc_idsminetd service

Use the following commands to manage the **tc_idsminetd** service.

- Restart the service if it is modified:

```
sudo systemctl daemon-reload
```

- Stop the service:

```
sudo systemctl stop tc_idsminetd.service
```

- View the service status:

```
sudo systemctl stop tc_idsminetd.service
```

- View the service journal entries:

```
sudo journalctl -u tc_idsminetd.service
```

- Disable the service on system startup:

```
sudo systemctl disable tc_idsminetd.service
```

Configure the IDSM on Windows systems

1. Edit the *TC_BIN***run_tc_idsm.bat** script file.

   Make these edits on both the proxy server and the client hosts.

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.

Configure proxy servers

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.

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. Enter the target directory for **TC_ROOT** in the installation directory box.

   > Caution:
   >
   > Do not select any other solution with the Proxy Server Solution.

   d. Enter 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 as follows.

   Because a proxy server does not use a database, it uses the **tc_preferences_overlay.xml** file to store Teamcenter 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.

     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.

Firewall configuration	Requirements
	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 preferences.

5. (Optional) For each client site, set an alternate proxy host by defining the following 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 preference is defined as:

   IDSM_proxy_client_alternate_proxy_host_for_
   myproxy1=myproxy2

Siemens Digital Industries 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.

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** preference. The preference must be set for each site to specify the port number used by clients to contact the remote ODS and IDSM servers.

Bypassing the RPC service requires firewall configurations for both the ODS and IDSM servers.

Remote procedure call security

You can configure or customize Multi-Site use of remote procedure calls (RPC) to meet your enterprise security policies.

By default, the Multi-Site RPC server uses random TCP ports over a range of ports. The range varies depending on your platform. If your enterprise security requires servers to communicate on static port numbers, you can customize the Multi-Site configuration to run the ODS and IDSM servers on specified port numbers.

Multi-Site remote import or remote export operations can be extremely long in duration depending on the size of volume data copied and the available bandwidth. The RPC connection is inactive during the volume data transfer. A firewall may close a connection that has been in a inactive state for too long. To avoid this issue, increase the time-out value of RPC connections on the firewall to permit the largest possible transactions to complete.

The **portmapper** service may be blocked by Internet-facing firewalls due to potential security issues. RPC clients normally require access to the **portmapper** service on the server site that runs on port 111 as defined in the RPC standards. On some platforms, the **portmapper** is called the **rpcbind** service.

You can bypass the portmapper service to avoid this security restriction.

Network topologies sometimes require the use of proxy servers. The most common use case in Multi-Site is the requirement to isolate internal network topography from client sites. Multi-Site supplies a custom proxy server configuration that supports both ODS and IDSM communication.

Configuring Multi-Site for HTTP/HTTPS communications

Configure Multi-Site authentication using HTTP/HTTPS

Prerequisites

Ensure the following conditions are met when configuring Multi-Site Collaboration to communicate using the HTTP/HTTPS protocol while sharing a common SSO domain.

- 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 must be configured to open ports for ODS and IDSM connections.)
- For HTTPS, enable Security Services single sign-on (SSO) functionality. All participating sites must be in the same SSO domain. Set the **TC_SSO_app_id_of_site_***site-name* preference at the site. *site-name* represents the name of the site.
- Ensure the **multisiteimport** element is properly configured in the primary configuration file for each site as described in **Configure FMS**.
- Ensure the following preferences are set:

Preference	Setting
TC_SSO_enabled	(Required for HTTPS.) Set to **true**.
TC_SSO_app_id_of_site_*site-name*	(Required for HTTPS.) The site's ID. *site-name* represents the name of the site. The preference value must match the Application ID value for the site as defined in the Application Registry table.
ODS_site	The default Object Directory Services (ODS) site.
ODS_permitted sites	Sites that can access the ODS database.
ODS_searchable_sites	ODS sites that Teamcenter searches for published remote objects during a remote search.
ISDM_permitted_sites	Sites that can access your data using the IDSM server.
IDSM_permitted_transfer_sites	Sites that are authorized to transfer ownership of objects owned by the site served by an IDSM server.
TC_transfer_area	The server directory used to temporarily storing data during import and export.

Create an HTTP/HTTPS enabled site

1. Select the top-level sites node  from the **Organization List** tree.

2. In the **Sites** pane, enter a descriptive name for the site for **Site Name**. For example, Teamcenter site 1 defines two remote sites: **TcHost2** and **TcEntHost**.

3. For **Site ID**, enter a unique identifier. For example, enter **457655709** for the **TcHost2** site.

   > Caution:
   >
   > Each site must be defined at other sites using exactly the same site ID in each definition.

4. For **Site Node/URL**, enter the URL used to contact the web application solution for the site. For example, if you deployed the web application that connects to the **TcHost2** site using default values for the application name and the application server listener port, enter **http://TcHost2:7001/tc**.

5. (Optional) If this is an ODS site, check **Provide Object Directory Services**.

6. (Optional) If this is a hub site, check **Is A Hub**.

7. Check **HTTP Enabled Multisite**.

8. (Optional) Check **Allow deletion of replicated master object to this site**. Selecting this allows deleting primary objects which have been replicated to the site, even if there is a replica existing for this primary object at the site.

9. Click **Create**. The site is created is listed in the **Organization List** tree.

10. When setting up HTTPS, continue by configuring Multi-Site Collaboration for HTTPS as follows.

Configure Multi-Site Collaboration for HTTPS

1. 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.

   If you are not using a vendor-generated certificate file, ensure your certificate file covers all sites in your Multi-Site Collaboration federation.

   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-----
```

2. Add the following properties to the *fsc.clientagnet.properties* file:

```
com.teamcenter.fms.curl.cacerts.file=certificate_file
com.teamcenter.fms.allowuntrustedcertificates=true
```

where *certificate_file* is the path and file name of SSL CA file.

Configure Multi-Site Collaboration for HTTPS using Deployment Center

Prerequisites

Install Teamcenter as described in Teamcenter Installation Using Deployment Center, ensuring the following items are configured appropriately. (Not all of these items can be configured using Deployment Center.)

- The site table entry for the remote site is created.
- A Multi-Site proxy user has been in the remote site's secondary LDAP server.
- Communication exists with the remote site 4-tier application server.
- The remote site is configured for FSC use.

Configure Multi-Site Collaboration for HTTPS

Run Deployment Center and ensure the following items are configured appropriately. Then, deploy the scripts created using Deployment Center.

1. On the **Components** tab, click **Add component to your environment** ⊕, check the **Multisite Collaboration HTTP** component, then click **Update Selected Components** to add it to the **Selected Components** list.

2. In the **Selected Components** list on the **Components** tab, select **Multisite Collaboration HTTP** and click **Show All Parameters** 👁.

3. Set the following values and click **Save Component Settings** when complete.

- Set **Machine Name** to the name of the machine on which Multi-Site Collaboration for HTTPS is being configured (the target machine).
- Set **OS** to the OS used on the target machine.
- Set **Teamcenter Installation Path** to the Teamcenter path installation on the target machine.
- Set the **Remote Site Credentials**, **Allowed remote credentials to local site**, and **Base Multisite preferences** values to the requested values.

Configure Multi-Site authentication using secondary LDAP servers configured with TcSS

Prerequisites

Ensure the following conditions are met:

- 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 must be configured to open ports for ODS and IDSM connections.)
- ApacheDS or another alternate secondary SSO (Single Sign-On) LDAP is required on each site. See Enable Teamcenter utilities to run with Security Services.
- A Teamcenter Multi-Site proxy user is required and must also exist in the secondary alternate LDAP directory.
  - Proxy users are not required to have Teamcenter administrator privileges. Because exports are performed by proxy users, proxy users must have read, export, and transfer out (for ownership transfer) privileges.
  - Proxy users may be unique for each site.
  - Sites may have multiple proxy users and may assign different users to support specific client sites.
- Ensure the **multisiteimport** element is properly configured in the primary configuration file for each site as described in **Configure FMS**.
- Network connectivity must be established from the client corporate server to the remote site web tier. For example, the Teamcenter server pool manager, **data_share** utility, and **data_sync** utility on the client corporate server must be able connect to the remote server.
- Ensure the following preferences are set:

Preference	Description
TC_use_alternate_sso	Lists all remote sites that are configured to use an alternate SSO LDAP server for HTTP Multi-Site authentication.
TC_use_alternate_sso_proxy_table	Lists remote proxy user authentication credentials for each site configured to use an alternate SSO LDAP server for HTTP Multi-Site authentication.
TC_alternate_sso_client_proxy_table	Server side preference that lists sites with HTTP Multi-Site access and the authorized proxy user for each site of the sites.
ODS_site	The default Object Directory Services (ODS) site.
ODS_permitted_sites	Sites that can access the ODS database.
ODS_searchable_sites	ODS sites that Teamcenter searches for published remote objects during a remote search.
IDSM_permitted_sites	Sites that can access your data via the IDSM server.
IDSM_permitted_transfer_sites	Sites that are authorized to transfer ownership of objects owned by the site served by an IDSM server.
TC_transfer_area	The server directory used to temporarily storing data during import and export.

Create an HTTP/HTTPS-enabled site

1. Select the top-level sites node  from the **Organization List** tree.

2. In the **Sites** pane, enter a descriptive name for the site for **Site Name**. For example, Teamcenter site 1 defines two remote sites: **TcHost2** and **TcEntHost**.

3. For **Site ID**, enter a unique identifier. For example, enter **457655709** for the **TcHost2** site.

   > Caution:
   >
   > Each site must be defined at other sites using exactly the same site ID in each definition.

4. For **Site Node/URL**, enter the URL used to contact the web application solution for the site. For example, if you deployed the web application that connects to the **TcHost2** site using default values for the application name and the application server listener port, enter **http://TcHost2:7001/tc**.

5. (Optional) If this is an ODS site, check **Provide Object Directory Services**.

6. (Optional) If this is a hub site, check **Is A Hub**.

7. Check **HTTP Enabled Multisite**.

8. (Optional) Check **Allow deletion of replicated master object to this site**. Selecting this allows deleting primary objects which have been replicated to the site, even if there is a replica existing for this primary object at the site.

9. Click **Create**. The site is created is listed in the **Organization List** tree.

10. When setting up HTTPS, continue by configuring Multi-Site Collaboration for HTTPS as follows.

Configure Multi-Site Collaboration for HTTPS

1. 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.

   If you are not using a vendor-generated certificate file, ensure your certificate file covers all sites in your Multi-Site Collaboration federation.

   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-----
   ```

2. Add the following properties to the *fsc.clientagnet.properties* file:

   ```
   com.teamcenter.fms.curl.cacerts.file=certificate_file
   com.teamcenter.fms.allowuntrustedcertificates=true
   ```

   where *certificate_file* is the path and file name of SSL CA file.

Installing and managing HTTP SSO proxy users

Proxy users are required to use SSO HTTP Multi-Site. A proxy user must be installed on each site. Use the **data_share** utility to manage proxy users as follows.

Use a command of the following form to install a proxy user for Multi-Site SSO:

```
urun data_share -f=install_http_proxy_user -connect_site=site1
-proxy_user=user1 -proxy_group=user1grp -proxy_role=user1role
-proxy_user_pwd=user1pw -u=tcadmin -p=tcpwd
```

Use a command of the following form to update the user at the site:

```
urun data_share -f=install_http_proxy_user -connect_site=site1
-proxy_user=newuser1 -proxy_group=new1grp -proxy_role=new1role
-proxy_user_pwd=newpw -u=tcadmin -p=tcpwd
```

Use a command of the following form to remove the user from the site:

```
urun data_share -f=remove_http_proxy_user -connect_site=site1 -u=tcadmin
-p=tcpwd
```

Example workflow

In this example, the following authentication occurs when Site1 sends an item to Site2 using the **data_share** utility to perform a remote export of the item.

1. Site1 checks its Site2 site object to verify it is HTTP enabled.

2. Site1 checks the value of the **TC_use_alternate_sso** preference to verify Site2 is configured to support an alternate SSO LDAP server for HTTP Multi-Site authentication.

3. Site1's **TC_alternate_sso_proxy_table** preference provides the credentials used to connect to Site2.

4. Site1 initiates an SOA call to Site2 using the proxy user credentials.

5. Site2 validates the credentials against its TcSS secondary LDAP.

6. Site2 checks the **TC_alternate_sso_client_proxy_table** preference to ensure this proxy user is valid for Site1.

Configure data_share for an HTTPS based Multi-Site environment

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.

Improve performance by redirecting Multi-Site HTTP related services to a dedicated server

When using Multi-Site with HTTPS, if you are using proxy servers with no authentication for outgoing and incoming traffic, there is no additional configuration needed. Multi-Site works with the existing infrastructure. If you are using authenticated proxy servers for either incoming or outgoing traffic or both, you must set up a secure channel (such as VPN) to allow Multi-Site traffic through the authenticating servers.

You can optionally enhance performance by setting up a dedicated web server and redirecting Multi-Site requests to the web server. The dedicated web server must be configured to block all requests other than the Multi-Site SOA calls. You can then configure you load balancer accordingly.

Following are the Multi-Site-related SOAs:

- Internal-MultiSite-2007-06-ObjectDirectory/*
- Internal-MultiSite-2011-06-ObjectDirectory/*
- Internal-MultiSite-2012-02-ObjectDirectory/*
- Internal-MultiSite-2007-06-RemoteOperation/*
- Internal-MultiSite-2012-02-RemoteOperation/*
- Core-2008-06-Session/loginSSO/*

Using HTTP enabled Multi-Site with forward proxy servers

Configure the rich client using 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.

> Note:
>
> Multi-Site uses FMS for file transfers. Therefore, you must configure Multi-Site for FMS whether you use a proxy server or not.

Adding a proxy server

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.

A user performing remote import operations through a proxy server can request automatic synchronization and notification.

Configuring preferences with a 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.

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*

Because 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.

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 workflow 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 workflow 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.

The IDSM user must have write access to a primary 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**.

Customizing an ODS schema

Adding attributes to the publication record

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.

Customizing attributes for a publication record requires the following:

- The client and server side must be customized to support the 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 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 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.

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 nonstring attributes

You implement custom exits to add nonstring custom attributes or attributes that are not part of the object POM attribute list or primary form. Custom 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 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 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** exit.
- The attribute names added to the local publication record.
- The attribute names added to the ODS server publication record.

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.

1. Access the **Advanced** perspective by choosing **Window→Open Perspective→Other→Advanced**.
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.

Configuring site-specific display rules

When deploying a custom template on one or more of the sites in your network, you can ensure that the custom template is always preferred over the common template as shown in the following approach for creating a custom type display rule:

1. Create a new type display rule in a site-specific template using a custom condition, for example **CustomDispType**.

2. Ensure that **CustomDispType** has a condition that simulates a behavior that is always true. Doing so ensures that the custom template is always preferred over the **isTrue** condition used by the type display rule in the common template. See Using conditions with naming rules.

3. Deploy your custom template.

See Creating, deploying, and packaging templates for more information.

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.

Be aware of the following items when customizing dataset export behavior:

- 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.

9. Setup verification

Post-setup checks

After you have Multi-Site Collaboration installed and configured check that you have your preferences, and before it is placed in operation use these setup checks to ensure a Multi-Site Collaboration system is properly installed and configured. This does not mean that the checks should be used only after the initial Multi-Site Collaboration configuration. The check can be used to diagnose a Multi-Site Collaboration problem when the system has been operational for some time or after a change in the configuration.

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.
```

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
```

Multi-Site deployment-related preferences

Depending on the deployment option your organization has chosen, the following preference settings are required.

RPC

Preference	Description
IDSM_permitted_sites	Sites that can access your data using the IDSM server.
IDSM_permitted_transfer_sites	Sites that are authorized to transfer ownership of objects owned by the site served by an IDSM server.
ODS_site	The default Object Directory Services (ODS) site.
ODS_permitted_sites	Sites that can access the ODS database.
ODS_publication_sites	ODS sites to use to publish objects.
ODS_searchable_sites	ODS sites that Teamcenter searches for published remote objects during a remote search.
TC_transfer_area	The server directory used to temporarily storing data during import and export.

Multi-Site using HTTP

Preference	Description
TC_SSO_enabled	Set to **true**.
TC_SSO_app_id_of_site_*site-name*	The TcSS SSO application id for every HTTP site in the federation.
IDSM_permitted_sites	Sites that can access your data using the IDSM server.
IDSM_permitted_transfer_sites	Sites that are authorized to transfer ownership of objects owned by the site served by an IDSM server.
ODS_site	The default Object Directory Services (ODS) site.
ODS_permitted_sites	Sites that can access the ODS database.
ODS_publication_sites	ODS sites to use to publish objects.
ODS_searchable_sites	ODS sites that Teamcenter searches for published remote objects during a remote search.
TC_transfer_area	The server directory used to temporarily storing data during import and export.

Multi-Site using secondary LDAP servers configured with TcSS

Preference	Description
TC_use_alternate_sso	Lists all remote sites that are configured to use an alternate SSO LDAP server for HTTP Multi-Site authentication.
TC_alternate_sso_proxy_table	Lists remote proxy user authentication credentials by site. For example, setting this preference to a value of **site2|***msproxy***|***group***|***role***|c:\secure\multisitepasswordfile** specifies that, when connecting to site2, use the

Preference	Description
	msproxy credentials with the password stored in the **| c:\secure\multisitepasswordfile** file to log on. This preference can be set by Teamcenter administrators only.
TC_alternate_sso_client_proxy_table	Server side preference that lists sites with HTTP Multi-Site access and the authorized proxy user for each site of the sites. For example, setting this preference on site2 to a value of **site1|***msproxy***|***group***|***role* specifies that site2 allows only the user with those specific *msproxy* can log on from site1. This preference can be set by Teamcenter administrators only.
IDSM_permitted_sites	Sites that can access your data via the IDSM server.
IDSM_permitted_transfer_sites	Sites that are authorized to transfer ownership of objects owned by the site served by an IDSM server.
ODS_site	The default Object Directory Services (ODS) site.
ODS_permitted_sites	Sites that can access the ODS database.
ODS_publication_sites	ODS sites to use to publish objects.
ODS_searchable_sites	ODS sites that Teamcenter searches for published remote objects during a remote search.
TC_transfer_area	The server directory used to temporarily storing data during import and export.

Operating system directories and files

Transfer area directory

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: (Linux only)

inetd.conf file

If this entry is missing, the IDSM daemon does not automatically start when a remote import request, or other IDSM request, is sent to the site.

The account used must have Teamcenter administration privileges and 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'

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.

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.

10. Managing Multi-Site Collaboration behavior

Role-based preferences

TC_bom_level_export

DESCRIPTION

Controls assembly export options: prevents the entire assembly from being exported; allows the entire assembly to be exported, but prevents transferring site ownership, or allows the entire assembly to be exported. This preference does not apply when running command-line utilities such as the **item_export** or the **data_share** utility.

VALID VALUES

0	Prevents a user from exporting an entire assembly by disabling the **Include Entire BOM** button in the **Import/Export Options** dialog box.
9999	Prevents a user from transferring site ownership of an entire assembly by disabling the **Include Entire BOM** button in the **Import/Export Options** dialog box when the **Transfer Ownership** button is selected.
Commented	Allows the entire assembly to be exported.

DEFAULT VALUES

Not applicable.

DEFAULT PROTECTION SCOPE

Role preference.

TC_dataset_vers_export

DESCRIPTION

Prevents a user from exporting all dataset versions (unless transferring site ownership) by disabling the **Include All Versions** button in the **Import/Export Options** dialog box. This role-based preference applies even when running command-line utilities such as the **item_export** or the **data_share** utility.

VALID VALUES

Accepts the value **FALSE**. If set to any other value, the system ignores the preference setting.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Role preference.

TC_modified_only_export

DESCRIPTION

Prevents a user from turning off the **Include Modified Objects Only** button in the **Import/Export Options** dialog box. This option usually results in better overall import/export efficiency. This preference does not apply when running command-line utilities such as the **item_export** or the **data_share** utility.

VALID VALUES

Accepts the value **TRUE**. If set to any other value, the system ignores the preference setting.

DEFAULT VALUES

Not applicable.

DEFAULT PROTECTION SCOPE

Role preference.

TC_ownership_export

DESCRIPTION

Prevents a user from transferring site ownership by disabling the **Transfer Ownership** button in the **Import/Export Options** dialog box.

VALID VALUES

Accepts the value **FALSE**. If set to any other value, the system ignores the preference setting.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Role preference.

TC_relation_export_hidden

DESCRIPTION

Prevents users in certain roles from seeing all available relation types in the **Import/Export Options** dialog box. This prevents users from including the hidden relation types for import/export operations. All hidden relation types are automatically excluded from the import/export operations.

> Note:
>
> The **OBJIO-send-target-objects** action handler does not support use of the **TC_relation_export_hidden** preference. The action handler fails when it runs the preference. As a workaround, use the **-exclude_relation** argument in the **OBJIO-send-target-objects** action handler.

VALID VALUES

A list of valid relation type names, one name per line. The database name, not the display name, of a relation type must be given. Invalid values in the list are ignored.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Role preference.

TC_sync_auto_synchronize

DESCRIPTION

Specifies whether Teamcenter automatically synchronizes replicas at remote sites when the primary object is modified. The **Synchronize automatically** check boxes in the **Remote Export Options** and **Remote Import Options** dialog boxes are not user selectable. Automatic synchronization is controlled only by the **TC_sync_auto_synchronize** preference. The check box value is read-only and provided for informational purposes.

Set the following values when using the **TC_sync_auto_synchronize** preference:

Setting	Value
Protection Scope	**Role** (Recommended)
Category	**Data Sharing.Multi-Site Collaboration** (Recommended)
Environment	**Disabled** (Recommended)
Type	**Logical**
Multiple	**Single**

VALID VALUES

One of any logical pair (**true** or **false**, **on** or **off**, **0** or **1**).

TRUE	The **Synchronize automatically** check box is selected in the **Remote Export Options** dialog box. Teamcenter performs automatic synchronization of remote objects.
FALSE	The **Synchronize automatically** check box is cleared in the **Remote Export Options** dialog box. Teamcenter does not perform automatic synchronization of remote objects.

DEFAULT VALUES

None.

You must create this preference manually. If the preference does not exist, the **Synchronize automatically** check box is cleared and Teamcenter does not perform automatic synchronization of remote objects.

DEFAULT PROTECTION SCOPE

Role preference.

Baseline_auto_remote_checkout_allowed

DESCRIPTION

Determines whether automatic remote checkout of replicated objects is allowed during a baseline operation. If allowed, replica items, and BOM views of replica item revisions, will be (automatically) checked out during baseline operations. The resulting baseline object is owned by the original owning site once the item or BOM view is remotely checked back in (automatically) after the baseline operation successfully completes.

Use this preference at sites using the remote check out/check in method, rather than the method of transferring ownership of objects. If your site often creates baselines of remote objects, enabling this preference improves system performance.

> Caution:
>
> Before setting this preference to **True**, ensure that the baseline business rules are the same at the owning site and remote site. Otherwise, revision IDs produced at the baseline will be of inconsistent values.
>
> For more information about setting business rules, see *BMIDE for Data Model Design*.

> Caution:
>
> The status type to be applied at the replica site must also exist at the owning site.

VALID VALUES

True or On or 1 (case insensitive)	Automatic remote checkout of replicated objects is allowed during baseline operations.
False or Off or 0 (case insensitive)	Automatic remote checkout of replicated objects is not allowed during baseline operations. Users can remote checkout replica objects manually, and proceed with a baseline operation.

DEFAULT VALUES

False

DEFAULT PROTECTION SCOPE

Site preference.

HUB_no_export_record_transfer_site_list

DESCRIPTION

Prevents the distribution of export records when site ownership is transferred from the hub. Use this preference to hide sensitive site information from unauthorized sites.

This preference is valid only for a hub site. When an export record is created at the importing site, the site objects are created if they do not already exist. As a result, information that may be confidential (such as the site ID, site name, and IP address) could be inadvertently distributed without this preference.

VALID VALUES

Accepts multiple strings as values. Each string must be a valid site authorized to transfer site ownership from the hub.

DEFAULT VALUES

None. If this preference is not defined, all export records are always be transferred to the new owning site of the data.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_do_not_resend_up_to_date_file

DESCRIPTION

Specifies whether Teamcenter must prevent the retransmission of a replicated file that is still current at the time an attempt to reimport or resend the file is made. Set this preference value to **TRUE** to prevent retransmission. This preference applies whether or not transfer of site ownership is involved. If no value is set for the preference, Teamcenter always retransmits the files.

> Warning:
>
> If this preference value is set to **TRUE** at a site, instruct the site's users *not* to delete replicas if they intend to reimport the object. This can cause new metadata but without the required file. If a missing file occurs, temporarily disable this preference, reimport the dataset, and then enable the preference.

VALID VALUES

TRUE	Prevents the retransmission of up-to-date files.
FALSE	Allows the retransmission of up-to-date files.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_dsa_sites_permitted_to_push_admin_data

DESCRIPTION

Defines the remote sites that are permitted to distribute system administration data to the local site. This enforces site-level security; if a remote site is not defined in this preference, that site cannot perform distributed system administration functions that affect the local site.

This is the first preference to be checked whenever a remote site tries to send system administration data to the local site. If this initial check results in a rejection of the request, the other distributed system administration preferences are not checked.

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_exclude_unpublished_components

DESCRIPTION

Prevents automatic replication of unpublished components when a published assembly is replicated.

This preference can be used only in conjunction with remote imports; it is not used when you choose **Tools→Export→Objects** from the My Teamcenter menu to export components.

VALID VALUES

TRUE	Unpublished components are not included when you perform a remote import of a published assembly and the **Include Entire BOM** command is enabled.
FALSE	All components—published and unpublished—are included.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_buffer_kb

DESCRIPTION

Sets the size of the network send and receive buffers in kilobytes. The maximum default is **64K** for traditional TCP implementations and is adequate for the majority of users.

Users with extremely high-performance networks may want to increase this setting. On some operating systems, you may need to enable the RFC1323 (Large Windows) TCP feature to increase buffers beyond the default.

Calculate the optimum buffer size by multiplying the bandwidth by the delay. Typically, the peak bandwidth of a network link is expressed in megabits per second (Mbit/s). For example, a T1 line is rated at 1.544 Mbit/s. The round-trip delay for a link can be measured with a trace route, and for WAN links is typically between 50 m/sec and 250 m/sec.

For example, a 200 m/sec delay, 2 Mbit/s path, the bandwidth delay product is 400 Kbit/s, or 50 Kbytes/s. In this example, for optimal performance, set the **IDSM_ft_buffer_kb** site preference to at least **50**.

VALID VALUES

Size of network send/receive butters in kilobytes.

DEFAULT VALUES

64K (Commented out)

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_client_timeout

DESCRIPTION

Sets the amount of time, in seconds, that the client waits for file transfer connections to respond before terminating the request.

VALID VALUES

A positive integer.

DEFAULT VALUES

30 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_get_port_number_from_services

DESCRIPTION

Determines where the IDSM retrieves the TCP/IP port number.

VALID VALUES

TRUE	Overrides the **IDSM_ft_port_number** preference setting and instructs the IDSM to look up the TCP/IP port number in the **/etc/services** file for the service named **idsmft**. On Windows, the file is located at **system32\drivers\etc\services**.
FALSE	Does not override the **IDSM_ft_port_number** preference setting.

DEFAULT VALUES

FALSE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_port_number

DESCRIPTION

Specifies the TCP/IP port number to use for file transfers. Must be different from the one in use for IDSM and ODS, and must not conflict with other registered system services.

Siemens Digital Industries Software recommends the default setting be changed to an unused port number in the reserved port range. Contact your network administrator for this information.

VALID VALUES

Valid port number.

DEFAULT VALUES

47953 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_server_timeout

DESCRIPTION

Sets the amount of time in seconds that the client and server wait for file transfer connections to respond before giving up.

VALID VALUES

A positive integer.

DEFAULT VALUES

30 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_ft_use_rpc_mode

DESCRIPTION

Determines which file transfer mode is used.

VALID VALUES

TRUE	Forces the use of the original Classic File Transfer mode. This mode is a slower remote procedure call (RPC) based transfer mode.
FALSE	Does not force the use of the original Classic File Transfer mode.

DEFAULT VALUES

FALSE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_global_dsa_set_local_volume_on_import

DESCRIPTION

Sets the local default volume for a user based on the name of the users default group name only for the initial import. Sets the local default volume for a group based on the group name only for the initial import.

VALID VALUES

Enter values as *Group-name*:*Volume-name*, one entry per line, for example:

```
Engineering:DrawingVol
```

DEFAULT VALUES

None

DEFAULT PROTECTION SCOPE

Site preference.

NOTES

This preference allows the importing site of users and groups to define what the local default volume is for initial import. The local default volume for a user is set by mapping its default group name to the volume name specified in the preference. The default local volume for a group is set by mapping the group name to the volume name specified in the preference.

RESTRICTIONS

- This preference is ignored for subsequent imports and synchronizations which preserve any existing value.
- This preference is not applicable for migration.

IDSM_global_dsa_set_volume_on_import

DESCRIPTION

Sets the default volume for a user based on the name of the users default group name only for the initial import. Sets the default volume for a group based on the group name only for the initial import.

VALID VALUES

Enter values as *Group-name*:*Volume-name*, one entry per line, for example:

```
Engineering:DrawingVol
```

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

NOTES

This preference allows the importing site of users and groups to define what the default volume is for initial import. The local volume for a user is set by mapping its default group name to the volume name specified in the preference. The default volume for a group is set by mapping the group name to the volume name specified in the preference.

RESTRICTIONS

- This preference is ignored for subsequent imports and synchronizations which preserve any existing value.
- This preference is not applicable for migration.

IDSM_global_dsa_sites_permitted_to_push_admin_data

DESCRIPTION

Defines the remote sites that are permitted to use Multi-Site export to push organization data to the local site. This enforces site-level security; if a remote site is not defined in this preference, that site cannot perform Multi-Site organization functions that affect the local site.

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_checkout_sites

DESCRIPTION

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

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_checkout_users_from_site_<sitename>

DESCRIPTION

Defines which user IDs from the sites specified by the **IDSM_permitted_checkout_sites** site preference are authorized to transfer ownership of objects owned by the local site.

If this preference is not defined, all users from a site defined by the **IDSM_permitted_checkout_sites** site preference can perform remote checkouts of objects owned by the local site.

Defining the **TC_check_remote_user_priv_from_sites** preference overrides this preference.

VALID VALUES

One or more strings; each string must be a valid Teamcenter user ID.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_sites

DESCRIPTION

Lists which sites can access your data via the IDSM server.

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_transfer_sites

DESCRIPTION

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.

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_transfer_users_from_site_<sitename>

DESCRIPTION

Defines which user IDs from the sites specified by the **IDSM_permitted_transfer_sites** site preference are authorized to transfer ownership of objects owned by the site served by an IDSM server.

Defining the **TC_check_remote_user_priv_from_sites** preference overrides this preference.

VALID VALUES

One or more strings; each string must be a valid Teamcenter user ID.

DEFAULT VALUES

None. If not defined, all users from the remote site are authorized to transfer data from the local site, provided the site is defined in the **IDSM_permitted_transfer_sites** site preference.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_permitted_users_from_site_<sitename>

DESCRIPTION

Defines which user IDs from the sites specified by the **IDSM_permitted_sites** site preference are authorized to replicate data owned by the site served by an IDSM server.

Defining the **TC_check_remote_user_priv_from_sites** preference overrides this preference.

VALID VALUES

One or more strings; each string must be a valid Teamcenter user ID.

DEFAULT VALUES

None. If not defined, all users from the remote site are authorized to replicate data from the local site, provided the site is defined in the **IDSM_permitted_sites** site preference.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_proxy_client_alternate_proxy_host_for

DESCRIPTION

Allows proxy clients to optionally specify an alternate proxy host. If the initial communication with the primary proxy host fails, the request is automatically sent to the alternate proxy host.

VALID VALUES

IDSM_proxy_client_alternate_proxy_host_for_*gmproxy1***=** *gmproxy2*

gmproxy1 is the regular proxy host node name and *gmproxy2* is the node name of the alternate proxy host.

Alternatively, an IP address can be used instead of the node name. The node name entry should match the entry for the node name in the site definition database. If the IP address is used in the database, the IP address should also be used in this preference

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_proxy_server_type

DESCRIPTION

Determines whether proxy server mode is available for the IDSM. A valid value enables the proxy server mode. When not set, the proxy server mode is disabled.

VALID VALUES

Relay	Enables relay proxy server mode.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

IDSM_restricted_sites

DESCRIPTION

Lists which sites cannot access your data via the IDSM server.

VALID VALUES

One or more strings; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_autopublish

DESCRIPTION

Enables or suppresses the automatic publication of all newly created items and the automatic republication of items when the item ID is modified. Newly created items are published to the ODS site specified by the **ITEM_autopublish_sites** site preference.

When an existing item ID is modified, it is republished only if it is already currently published and then only to the ODS sites to which it has already been published. The **ITEM_autopublish_sites** site preference does not determine which ODS site to use for republication.

VALID VALUES

TRUE	Enables the automatic publication of all newly created items and the automatic republication of items when the item ID is modified.
FALSE	Suppresses the automatic publication of all newly created items and the automatic republication of items when the item ID is modified.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_autopublish_ignore_errors

DESCRIPTION

Determines whether to ignore publication errors detected during automatic publication. When a publication error is detected, any item creations/modifications known to be successful at that point are retained; failed items can be published manually at your convenience. This preference requires the **ITEM_autopublish** site preference to be set.

Errors caused by invalid ODS site preference settings are ignored by this preference.

> Note:
>
> Errors due to item creation or modification are not covered by this preference; the preference applies only to publication errors.

VALID VALUES

TRUE	When automatic publishing is enabled via the **ITEM_autopublish** site preference, ignores publication errors detected during automatic publication.
FALSE	When automatic publishing is enabled via the **ITEM_autopublish** site preference, publication errors prompt the system to cancel the item creation or modification.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_autopublish_sites

DESCRIPTION

Defines the ODS site to which newly created items are published. This preference requires the **ITEM_autopublish** site preference to be set.

VALID VALUES

Accepts a single string as a value. Must be a valid ODS site.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_id_allow_if_registry_down

DESCRIPTION

Determines if items can be created if the registry server is not available. If the registry is not active, this preference is ignored.

VALID VALUES

TRUE	Items can be created when the item ID registry server is unavailable.
FALSE	Items cannot be created when the item ID registry server is unavailable.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_id_always_register_on_creation

DESCRIPTION

Determines if item IDs are automatically registered when items are created or the item ID is changed.

VALID VALUES

TRUE	Automatically registers item IDs to the item ID registry when items are created or the item ID is changed.
FALSE	Does not register item IDs to the item ID registry when items are created or the item ID is changed.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_id_registry

DESCRIPTION

Activates the item ID registry.

VALID VALUES

TRUE	During item creation (immediately before the new item is saved to the database), the system checks the item ID registry for duplicates. Creating the item fails if a duplicate is found.
FALSE	The item ID registry is not checked for duplicates while creating a new item.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_id_registry_site

DESCRIPTION

Identifies the site name upon which the item ID registry server is running.

This preference must be set if the item ID registry is active. If the registry is not active, this preference is ignored.

VALID VALUES

A single string as a value; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_id_unregister_on_delete

DESCRIPTION

Determines if item IDs are automatically unregistered when items are deleted or the item ID is changed.

VALID VALUES

TRUE	Automatically removes item IDs from the item ID registry when items are deleted or the item ID is changed.
FALSE	Does not remove item IDs to the item ID registry when items are deleted or the item ID is changed.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

ITEM_relation_types_update_lmd

DESCRIPTION

Determines which relation types cause the update of the last modification date of the parent object when the secondary object of a class dataset or form is modified, or if the relation is added/removed from the parent object. If the relation is added/removed, the last modification date of the parent object is updated regardless of the class of the secondary object.

VALID VALUES

Accepts one or more strings as values. Each string must be a valid Teamcenter relation. Excepting the **IMAN_RES_audit** and **IMAN_RES_checkout** relations, which are *not* valid values for this preference.

DEFAULT VALUES

IMAN_master_form
IMAN_specification
IMAN_requirement

The **IMAN_master_form**, **IMAN_specification** and **IMAN_requirement** relation types are implied entries in this preference and cannot be excluded.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_multiprocess_idle_subprocess_timeout

DESCRIPTION

Defines the length of time a multiprocess subprocess can be idle before it is taken out of service and terminated.

Used when the Object Directory Services (ODS) is running in multiprocess mode. In this mode, the ODS maintains a minimum number of subprocesses to respond to client requests. Additional subprocesses are created on demand, up to a maximum number (defined by the **ODS_multiprocess_max_subprocess_count** preference) when the number of concurrent outstanding client requests exhaust the capacity of the minimum number of subprocesses.

This preference's idle timeout setting applies to the subprocesses created on demand. The timeout is specified in minutes.

VALID VALUES

Any positive integer.

DEFAULT VALUES

10 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

ODS_multiprocess_initial_subprocess_count

DESCRIPTION

Determines the number of subprocesses created and logged into the database by the parent Object Directory Services (ODS) process at startup.

If too few subprocesses are created, remote users may notice a delay due to the time required to create a subprocess and log on to the database. If too many subprocesses are created, strain may be placed on system resources, resulting in slow performance.

VALID VALUES

Any positive integer.

DEFAULT VALUES

5 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

ODS_multiprocess_max_subprocess_count

DESCRIPTION

Determines the maximum number of precreated subprocesses that can be created by the parent Object Directory Services (ODS) process.

If the maximum number is too low, remote users may experience noticeable delays when performing ODS-related operations. If the maximum number is too high, too many system processes could be created, thus straining system resources and causing slow performance.

VALID VALUES

Any positive integer.

DEFAULT VALUES

10 (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

ODS_multiprocess_mode

DESCRIPTION

Enables or suppresses multiprocess Object Directory Services (ODS) operations, providing improved service when the ODS server process is being monopolized by remote operations such as report generation and object publication. Multiprocess mode can also be used to improve performance when service from the ODS server is slow due to high numbers of operations running at multiple remote sites.

VALID VALUES

TRUE	Enables multiprocess Object Directory Services (ODS) operations.
FALSE	Suppresses multiprocess Object Directory Services (ODS) operations.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

All.

ODS_permitted_sites

DESCRIPTION

Sets which sites can access the Object Directory Services (ODS) database.

VALID VALUES

One or more strings as values; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_proxy_server_type

DESCRIPTION

Determines whether proxy server mode is available for the ODS. A valid value enables the proxy server mode. When not set, the proxy server mode is disabled.

VALID VALUES

Relay	Enables relay proxy server mode.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_publication_sites

DESCRIPTION

Lists Object Directory Services (ODS) sites that can be used to publish an object simultaneously. The list serves as the recommended ODS list when publishing to multiple sites using **Commands→Publish→To ODS Publication List** menu commands.

This list plus the default ODS comprise the list of authorized ODS sites to which users can publish.

VALID VALUES

One or more strings as values; each string must be a valid Teamcenter site name and must be designated as providing ODS.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_restricted_sites

DESCRIPTION

Sets which sites cannot access the Object Directory Services (ODS) database.

VALID VALUES

One or more strings as values; each string must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_searchable_sites

DESCRIPTION

Specifies the ODS sites that Teamcenter searches for published remote objects during a remote search.

The sites set in this preference are listed in the **Shown** column for the **ODS Searchable Sites** section in the **Options** dialog box.

Nonadministrator users can create an instance at the user location in the **Details** pane of the **Options** dialog and add or remove values to override any site location settings.

VALID VALUES

One or more strings as values; each string must be a valid Teamcenter site name as defined in the Organization application.

RESTRICTIONS

Values that appear in **ODS_searchable_sites** preference for the site location are mutually exclusive from the values in **ODS_searchable_sites_excluded** preference for the site location.

Values that appear in **ODS_searchable_sites** preference for a user location are mutually exclusive from the values in **ODS_searchable_sites_excluded** preference for that user location.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

User preference.

ODS_searchable_sites_excluded

DESCRIPTION

Specifies the sites that Teamcenter does not include in searches for published remote objects during a remote search.

The sites set in this preference are listed in the **Shown** column for the **ODS Searchable Sites Excluded** section in the **Options** dialog box.

Nonadministrator users can create an instance at the user location in the **Details** pane of the **Options** dialog and add or remove values to override any site location settings.

VALID VALUES

One or more strings as values; each string must be a valid Teamcenter site name as defined in the Organization application.

RESTRICTIONS

Values that appear in **ODS_searchable_sites_excluded** preference for the site location are mutually exclusive from the values in **ODS_searchable_sites** preference for the site location.

Values that appear in **ODS_searchable_sites_excluded** preference for a user location are mutually exclusive from the values in **ODS_searchable_sites** preference for that user location.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

User preference.

ODS_site

DESCRIPTION

Sets the default Object Directory Services (ODS) site.

VALID VALUES

A single string as a value; must be a valid Teamcenter site name and must be designated as the providing ODS site.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

ODS_suppress_pubrec_if_no_access

DESCRIPTION

Prevents users at remote sites from knowing about publication records to which their site has no access, and is used only at Object Directory Services (ODS) sites.

This preference is used only at ODS sites.

For information about ODS sites, see *Multi-Site Collaboration*.

VALID VALUES

TRUE	Protected publication records are invisible, (that is, they do not exist for those users that do not have access). For remote sites running pre-V6.0 software, protected publication records are automatically invisible.
FALSE	Users at remote sites that do not have read access to a publication record see an **Access Denied** message in the object list window. **Find Remote** produces this message.

DEFAULT VALUES

FALSE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_<daemon-name>_<site-id>_prog_number

DESCRIPTION

Determines which custom remote procedure call (RPC) program number the IDSM or ODS server runs on the specified site name. This preference is only for use with a proxy configuration.

> Note:
>
> The IDSM or ODS server must first be properly configured to run on the custom RPC program number.

VALID VALUES

The syntax for this preference is **TC_***daemon-name*_*site-id*_**prog_number** where the *daemon name* is either the ODS or IDSM process (specified in lowercase), the *site-id* is the ID of a valid Multi-Site Collaboration site, and the **prog_number** is a valid open network computing (ONC) RPC number. For example:

```
TC_ods_1122334455_prog_number=
536875580
TC_idsm_1122334455_prog_number=
536875590
```

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_<daemon-name>_<site-name>_port_number

DESCRIPTION

Allows you to bypass the remote procedure call (RPC) portmapper service (**rpcbind**). You may want to avoid using the portmapper service for security reasons. The portmapper bypass configuration allows you to configure the ODS and IDSM servers to run on specific port numbers.

VALID VALUES

The syntax for this preference is **TC_***daemon-name*_*site-name*_**port_number** where the *daemon name* is either the ODS or IDSM process (specified in lowercase), and the *site name* is the name of a valid Multi-Site Collaboration site. For example:

```
TC_ods_athens_port_number=
47000
TC_idsm-athens_port_number=
47001
TC_ods_rome_port_number=
57000
TC_idsm_rome_port_number=
57001
```

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_<daemon-name>_<site-name>_prog_number

DESCRIPTION

Determines which custom remote procedure call (RPC) program number the IDSM or ODS server runs on the specified site name. This preference is not valid for a proxy configuration.

> Note:
>
> The IDSM or ODS server must first be properly configured to run on the custom RPC program number.

VALID VALUES

The syntax for this preference is **TC_***daemon-name*_*site-name*_**prog_number** where the *daemon name* is either the ODS or IDSM process (specified in lowercase), the *site name* is the name of a valid Multi-Site Collaboration site, and the **prog_number** is a valid open network computing (ONC) RPC number. For example:

```
TC_ods_athens_prog_number=
536875580
TC_idsm_athens_prog_number=
536875590
```

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

Tc_allow_users_edit_pfdata_options

DESCRIPTION

Determines whether options for publishing part family member items and templates can be modified by general users in the **Options** dialog box, accessed from the **Edit** menu. Users with system administrator privileges can always edit these options from the **Options** dialog box, regardless of how this preference is set.

VALID VALUES

TRUE	General users can modify options for publishing part family member items and templates using the **Options** dialog box.
FALSE	General users cannot modify options for publishing part family member items and templates using the **Options** dialog box.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_altrep_update_lmd

DESCRIPTION

Determines whether the last modified date of a locally owned item revision is updated when an object with an associated altrep relation is modified.

The last modified date of the parent item is automatically updated as a consequence of the associated item revision update.

VALID VALUES

TRUE	The last modified date of the item revision is modified when an object with an associated altrep relation is modified.
FALSE	The last modified date of the item revision is not modified when an object with an associated altrep relation is modified.

DEFAULT VALUES

TRUE

DEFAULT PROTECTION SCOPE

Site preference.

TC_always_exclude_dataset_files_on_export

DESCRIPTION

Determines whether the dataset named reference file is excluded from export. Use this preference in conjunction with the on-demand retrieval of dataset volume files functionality. This preference affects Multi-Site transfers only.

For more information about this functionality, see *Multi-Site Collaboration*.

VALID VALUES

true	Excludes the dataset named reference file from export. This ensures a given role can never store replica volume data at a remote site. This setting forces the remote user to retrieve the volume data from the owning site.
false	Does not exclude the dataset named reference file from export.

DEFAULT VALUES

Not applicable.

DEFAULT PROTECTION SCOPE

Site preference.

TC_appref_registry_site

DESCRIPTION

Defines which Object Directory Services (ODS) site is used for all Application Reference ITK operations. For use with Teamcenter Integration for NX I-deas migration. When I-deas data is migrated to Teamcenter, these APIs are used to ensure duplicate GUIDs are not created.

For more information about the Application Reference ITK module, see the *Integration Toolkit Function Reference*.

Use this preference to direct all API operations to a site other than the default ODS site (defined by the **ODS_site** site preference). This allows a separate ODS, dedicated to item GUIDs, to be setup independent of a regular ODS.

VALID VALUES

A single string as a value; must be a valid Teamcenter site name.

DEFAULT VALUES

The value defined in the **ODS_site** site preference.

DEFAULT PROTECTION SCOPE

Site preference.

TC_assembly_xml_relation

DESCRIPTION

Indicates the relation name that is used to relate the PLM XML data generated for the assembly.

Modify this preference value if you must have the PLM XML dataset attached using a specific relation for business needs.

VALID VALUES

Any existing relation name.

DEFAULT VALUES

IMAN_Specification

DEFAULT PROTECTION SCOPE

Site preference.

TC_background_object_export_dir

DESCRIPTION

Defines the directory where export data is stored when a background export is performed using the rich client.

When an interactive, non-remote background export operation is performed using the rich client, the export data is stored in a subdirectory of the defined directory; the subdirectory name is defined by the user in the **Object Export** dialog box. For example, if the user specifies that the export data be stored in the directory named *my_item100*, the background export output is placed in the following directory:

```
/mydisk/mydir/my_item100
```

VALID VALUES

Valid disk drive and directory name.

DEFAULT VALUES

c:\background_expdir (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_bom_level_export

DESCRIPTION

Controls assembly export options: prevents the entire assembly from being exported; allows the entire assembly to be exported, but prevents transferring site ownership, or allows the entire assembly to be exported. This preference does not apply when running command-line utilities such as the **item_export** or the **data_share** utility.

VALID VALUES

0	Prevents a user from exporting an entire assembly by disabling the **Include Entire BOM** button in the **Import/Export Options** dialog box.
9999	Prevents a user from transferring site ownership of an entire assembly by disabling the **Include Entire BOM** button in the **Import/Export Options** dialog box when the **Transfer Ownership** button is selected.
Commented	Allows the entire assembly to be exported.

DEFAULT VALUES

Not applicable.

DEFAULT PROTECTION SCOPE

User preference.

> Note:
>
> This value is generated by the system and should not be changed.

TC_check_owner_on_import

DESCRIPTION

Checks the owning user and the owning group of the primary object during a Multi-Site import operation. If the owning user or owning group of the primary object is not defined at the import site, an error is generated. It allows you to enforce a rule that the owning user and group of an object at the owning site must also exist at the importing site.

For example, on site 1, the item **Car** is owned by user **John** who is a member of the **Engineering** group. Performing an import at site 2 succeeds only if both user **John** and the **Engineering** group both exist at site 2.

VALID VALUES

Accepts a logical value such as **True** or **False**, **On** or **Off**, or **1** or **0**.

DEFAULT VALUES

False.

DEFAULT PROTECTION SCOPE

Site preference.

TC_check_remote_user_priv_from_sites

DESCRIPTION

Determines whether individual remote user privileges are checked before remote users are allowed to perform the remote operations listed below. Use this preference to list the remote sites included in this security check. Whenever remote operations are attempted from any remote site named in this preference, the system evaluates current AM rules for the IDSM site for the remote user ID. The security check is implemented on the IDSM server and applies to any remote site named by the local site.

If this preference is not set, the system applies AM rules only at the site level. In this situation, only the remote *site's* AM privileges are checked against the AM rule tree of the owning site; access to individual objects at the owning site are not validated against individual remote user privileges. In this scenario, the **IDSM_permitted_users_from_site_<sitename>**, **IDSM_permitted_transfer_users_from_site_<sitename>** and **IDSM_permitted_checkout_users_from_site_<sitename>** preferences can be used to determine remote user access to replicate, transfer, and to check in or to check out remote data. You will override these three preferences if you set the **TC_check_remote_user_priv_from_sites** preference.

The remote operations affected are:

- 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

VALID VALUES

Accepts one or more strings as values. Each string must be the name of a valid remote site.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_cms_relation_optset_map

DESCRIPTION

Controls mapping of Teamcenter relation names to the TC XML transfer option sets. Set this preference only when you want to control the relations that are included or excluded when an object is replicated.

Multi-Site Collaboration allows you to include or exclude relations during transfers. During TC XML transfers, Multi-Site maps the user supplied relations to the corresponding TC XML transfer option set entry using the name-value pairs set by this preference.

Use this preference to include or exclude relation objects during **data_share** or **data_sync** utility transactions that use the **-low_level** argument (TC XML transfers).

VALID VALUES

One or more name-value pair strings, for example:

```
IMAN_rendering,opt_rel_rendering
```

The name must be a valid Teamcenter relation, and the value must be a valid TC XML transfer options set entry.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_directly_transferable_classes

DESCRIPTION

Prevents any attachment of a replicated item from being transferred independent of the item. This prevent users replicating items containing attachments, then remote importing the attachments with a transfer of ownerships which prevents other users from further replicating the item because ownership of the item is no longer consistent. This situation can also cause synchronization problems.

To prevent direct transfer of ownership of all classes of objects except items and item revisions, set this preference to **Item**. To enable direct transfer of other classes, such as **Folder**, add the **Folder** class as a value.

VALID VALUES

Accepts one or more strings as values; each string must be a valid Teamcenter object class.

DEFAULT VALUES

Item
ItemRevision
Dataset
Folder
Form
PSBOMView
PSBOMViewRevision
EPMTaskTemplate

DEFAULT PROTECTION SCOPE

Site preference.

TC_disallow_release_status_on_replica

DESCRIPTION

Specifies whether replicas are allowed to be released (have a release status).

VALID VALUES

true	Prevents replicas from being released.
false	Allows replicas to be released (have a release status).

DEFAULT VALUE

true

DEFAULT PROTECTION SCOPE

Site preference.

TC_disallowed_replica_relations

DESCRIPTION

Prevents adding local attachments to replicas. Without this preference, relation types other than **IMAN_specification**, **IMAN_requirement** and **IMAN_master_form** can be used to attach files to replica items and item revisions. This makes it possible for users to attach important data to the replica even though the attachment is strictly local data that cannot be synchronized to other sites. This can be unacceptable, especially when the attachments are UG Altreps.

Use this preference to prevent the attachment of manifestations, references, and UG Altreps to replica items and item revisions.

VALID VALUES

IMAN_manifestation	Prevents the attachment of manifestations to replica items and item revisions.
IMAN_reference	Prevents the attachment of references to replica items and item revisions.
IMAN_UG_altrep	Prevents the attachment of UG Altreps to replica items and item revisions.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_do_not_define_sites_on_import

DESCRIPTION

Specifies whether or not sites that are not defined in an importing sites database are automatically defined when objects are imported to the site from a hub site.

Use this preference to prevent sites that are unknown to each other, such as supplier sites, from being automatically defined in the importing site's database, for example:

- Supplier sites **ABC** and **XYZ** exchange data through hub site **HUB**. The hub site is defined in both site **ABC**'s and site **XYZ**'s database. Site **ABC** is not defined in site **XYZ**'s database and site **XYZ** is not defined in site **ABC**'s database because they are not known to each other.

- Site **ABC** exports a replica object to the hub site. Site **XYZ** imports site **ABC**'s replica object from the hub site. By default, the import process defines site **ABC** in site **XYZ**'s database making the site information visible to the site **XYZ** users.

- Define this preference at site **XYZ** with site **ABC** as the value. This prevents site **ABC** from being defined in site **XYZ**'s database during the import process and prevents users at site **XYZ** from seeing information about site **ABC**. Do the same for site **XYZ** at site **ABC** to prevent uses at either site from knowing about the other site.

VALID VALUES

One or more strings, for example:

```
SupplierABC
```

The name must be a valid Teamcenter site name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_do_not_insert_imported_objects_into_folder

DESCRIPTION

Determines whether imported objects are added to the user's home folder. This preference affects imported objects pushed to a specific user at a remote site and objects imported by the user using the **item_import** utility.

In situations where the home folder contains thousands of objects, set this preference to **TRUE** to improve performance.

VALID VALUES

TRUE	Imported objects are not inserted into the specific user's home folder. Objects are stored in the database with no reference to any folder. To locate the imported objects, users must search the database.
FALSE	Imported objects are inserted into the specific user's home folder.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_EMAIL_AFTER_BACKGROUND_REMOTE_IMPORT

DESCRIPTION

Controls whether an e-mail notification is sent after completion of a background remote import operation.

The e-mail address must be defined in the person object of the user who started the remote import. Otherwise, an e-mail is not sent.

VALID VALUES

TRUE	An e-mail notification is sent after completion of a background remote import operation.
FALSE	An e-mail notification is not sent after completion of a background remote import operation.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

Tc_export_pfmember_replica

DESCRIPTION

Determines whether part family member replicas can be exported, or remote imported, from a remote site to another remote site. This preference applies when exporting or remote importing assemblies containing the part family member replica components.

VALID VALUES

TRUE	Allows the export of part family member replica components.
FALSE	Does not allow the export of part family member replica components.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_follow_ownership_chain_max_site_count

DESCRIPTION

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**.

VALID VALUES

You may use any integer value up to the number of known sites.

No limit	Specifies Multi-Site shall continue to query for the replicas owner until all known sites are queried.

DEFAULT VALUES

No limit

DEFAULT PROTECTION SCOPE

Site preference.

TC_force_remote_sites_exclude_files

DESCRIPTION

Determines if the exporting site sends replica files to the file server cache (FSC) or stores it in the volume. The exporting site interrogates the importing site for the this preference value. Therefore, this preference impacts the exporting site only for online cases.

VALID VALUES

TRUE	Stores the replica files in the FSC.
FALSE	Stores the replica files in the volume.

DEFAULT VALUE

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

NOTES

- The **TC_force_remote_sites_exclude_files** preference does not override the display of any options in the rich client. It is a silent override.
- This preference must be set at the importing site.

TC_hub_groups

DESCRIPTION

Defines replication behavior for sharing data via Multi-Site Collaboration hubs. The replication behavior is managed by assigning hub clients (sites connected to a hub for data sharing purposes) to arbitrary hub groups.

The following rules govern behavior of hub group memberships. Given the following example:

TC_hub_groups=
Company 1:partner
Company 2:partner
Company 3:supplier
Company 4:supplier

- When the site importing from the hub, and the real owning site of the replica being imported from the hub, are in the same hub group, then the requesting site receives its copy directly from the real owning site. In the above example, assume Company 1 sends a replica of Item 1 to the hub. If Company 2 replicates Item 1 from the hub, the hub does not send Item 1 to Company 2, but redirects the request to Company 1 because Company 1 and Company 2 are in the same hub group.

  This behavior applies to components when importing a whole assembly. However, for Company 2 to automatically receive components owned by Company 1, the **Include Distributed Components** import/export command must be selected; otherwise, the components owned by Company 1 are stubbed at Company 2.

- When the site importing from the hub and the real owning site of the replica being imported from the hub are in different hub groups, then the requesting site receives its copy from the hub (which is normal hub replication behavior). In the above example, assume that Company 1 sends a replica of Item 1 to the hub. If Company 3 tries to replicate Item 1 from the hub, the hub sends a replica of Item 1 to Company 3 because Company 1 and Company 3 are in different hub groups.

- When publishing a replica that resides on the hub when the owning site of the replica and the target ODS are in the same hub group, the publication is not allowed. If the owning site and the target ODS are in different hub groups, the publication is allowed.

  In the above example, assume each site has its own ODS that is accessible from the hub. Also assume that Company 1 sends a replica of Item 1 to the hub. If a user logged onto the hub tries to publish Item 1 to the ODS of Company 2 the publication is not allowed because Company 1 and Company 2 are in the same hub group. Presumably, Company 1 and Company 2 are already sharing data directly with each other and there is no need to share replicas via the hub. Alternatively, if the user is logged onto the hub and tries to publish Item 1 to the ODS of Company 3, then the publication is allowed. This enables Company 1 to search Company 3's ODS and still receive its replicas from the hub and not from Company 3.

VALID VALUES

Accepts one or more strings as values. Define the preference only in the hub database, using the following format:

TC_hub_groups=
site1:*hub group1*
site2:*hub group2*
site3:*hub group3*

Where *site* is the name of the hub client site. Do not enter the hub site itself as a value.

And where *hub group* is a user-assigned name (maximum 32 characters) used for grouping client sites; such customer-defined grouping is the suggested method of managing replication and publication behavior for data shared via Multi-Site Collaboration hubs.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_identify_plmxml_import_dataset

DESCRIPTION

Specifies the relation name, dataset type, and the named reference that identifies the dataset used to parse the named reference PLM XML file and identify the item revisions to be imported.

VALID VALUES

> Note:
>
> Although this preference accepts multiple string values, this to allow importing any PLM XML file. For use with controlled replication of structured context objects, this preference must be a single string containing the dataset type, relation name, and named reference type.

Multiple string values delimited by colons (:) representing any existing dataset type, relation name, and named reference type in the following format:

```
<dataset-type>:<relation>:<named-reference–type>
```

If a named reference type is not specified, **ConfiguredAssembly** is used.

DEFAULT VALUES

DirectModelAssembly:IMAN_reference:ConfiguredAssembly

DEFAULT PROTECTION SCOPE

Site preference.

RESTRICTIONS

All values in the preference must exist in the Teamcenter database.

NOTES

Modify this preference value when the attached PLM XMLL dataset must be processed for import using a specific relation, dataset type, and named reference for your business needs.

TC_idsm_client_def_timeout

DESCRIPTION

Sets the basic timeout interval, in seconds, before network nodes (clients) attempting to connect to the IDSM terminate the request. This timeout value is used for all subsequent IDSM requests.

VALID VALUES

Single numerical value. Siemens Digital Industries Software recommends **300**.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_idsm_client_initial_timeout

DESCRIPTION

Sets the basic timeout interval, in seconds, before network nodes (clients) attempting to connect to the IDSM terminate the request. This timeout value is used for the initial request to the IDSM; this may take longer than subsequent requests.

VALID VALUES

Single numerical value. Siemens Digital Industries Software recommends **300**.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_idsm_proxy_server_site_table

DESCRIPTION

Maps the target site ID with its real node name so that the proxy server can redirect a requesting site's first remote procedure call (RPC) message to the target site.

VALID VALUES

This preference must list all sites (both client sites and server sites) that use the proxy host. This list provides the information needed to map site IDs to their real node names. The list is also used to discriminate against clients that are not authorized to use the proxy host. The proxy denies access to any requesting site not on the site table list. For example:

```
TC_idsm_proxy_server_site_table=
Site 1 ID:real-node-name-for-Site 1
Site 2 ID:real-node-name-for-Site 2
```

Site ID is the site ID of either an internal or external site that uses the proxy host and *real-node-name-for-Site* is the actual node name for the site ID.

The IP address can be used instead of the node name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_ie_error_report

DESCRIPTION

Determines whether a failure report is created. This preference is automatically set by the system when users select the **Save Options As Default** option in the **Synchronization Preferences** dialog box.

The dialog box allows users to perform on-demand synchronization (as opposed to administrative-based synchronization using utilities, or automatic synchronization which only occurs when the primary object is modified.) This functionality provides users with immediate visual confirmation that the synchronization has succeeded or failed. Access the dialog box by choosing **Multi-Site Collaboration→Synchronize** from the **Tools** menu, or right-clicking on object and choosing **Multi-Site Synchronization** from the shortcut menu.

VALID VALUES

TRUE	A failure report is generated.
FALSE	A failure report is not generated.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

User preference.

TC_master_locale_<site_name>

DESCRIPTION

Specifies the primary language for the given site defined in the site pane of the Organization application. This language is used as the locale for exports to the site or as the locale when importing from the specified remote site to the local site.

This preference must be set for all monolingual sites in a Multi-Site environment that also contains multilingual sites.

VALID VALUES

Accepts the standard Java language codes consisting of a two-character language string and a two-character country string separated by an underscore character. Each string must be one of the following codes indicating the desired locale:

cs_CZ
de_DE
en_US
es_ES
fr_FR
it_IT
ja_JP
ko_KR
ru_RU
zh_CN
zh_TW

DEFAULT VALUE

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_Multisite_tcxml_TOS_Mapping

DESCRIPTION

Provides a map of the export site transfer option set (TOS) to the import site TOS that Multi-Site uses for TC XML transfers. The preference allows execution of postactions registered in the import transfer mode for Multi-Site TC XML-based imports. Postactions allow you to set up utilities, such as the **appmodel_fix_scope** utility, to run automatically after a low-level TC XML import.

To use this preference, the import postaction must be specified as the action rule in the import site TOS.

VALID VALUES

One or more pairs of comma-delimited strings, for example:

```
ExportingSiteOptionSet,ImportingSiteOptionSet
```

DEFAULT VALUES

MultiSiteExpOptSet,MultiSiteImpOptSet

DEFAULT PROTECTION SCOPE

Site preference.

TC_not_directly_exportable_classes

DESCRIPTION

Specifies a list of classes that are prohibited from being directly selected as the top-level object in a Multi-Site export operation. For example, add **Form** and **Dataset** values to this preference if you want to block form and dataset objects from being directly exported by themselves, but allow export when they are part of an item or item revision.

If no values are specified, all classes are directly exportable.

VALID VALUES

Accepts one or more strings as values. Each string must be the name of a valid Teamcenter object.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_ods_client_extra_attributes

DESCRIPTION

Specifies and describes extra attributes of an object being published that you want stored in the publication record. Only string attributes can be specified. The attribute must have an initial value.

When adding custom attributes to the publication record, you may want to populate these extra attributes when an object is published. For example, a new **security level** attribute is added to the publication record and you want to specify where the publication code retrieves the security information associated with the object being published.

Use this preference for attributes that are part of the object's extended attributes (or any attributes from the primary form, if the object is an item or item revision).

In multifield key (MFK) environments, this preference must be defined on the ODS client side and must contain the required publication record key attributes.

> Note:
>
> When the custom attribute is not part of the object or its primary form, you must implement a user exit to populate the custom publication record attributes.

VALID VALUES

Accepts multiple strings as values. Each string must be in the following format:

class-name::*object-POM-attribute*:*pub-record-POM-attribute*:*option*

class-name	Specifies the name of the class that defines the **po_pom_attr_name** attribute. This is required only if **po_pom_attr_name** is used in multiple classes, in which case you must qualify the attribute name. This entry is optional and is normally not used.
object-POM-attribute	Specifies the POM name of the extra non-array string attribute of the object being published. This should either be part of the class definition of the object or part of its primary form (for items and item revisions). This entry is required.
pub-record-POM-attribute	Specifies the POM name of the extra non-array string attribute of the publication record class where the value of **po_pom_attr_name** is stored. This should be part of the publication record class definition at both client and server sides. This entry is required.

> Note:
>
> This attribute name cannot start with **pubr**, which is reserved for internal use.

option — Indicates whether an extra attribute is mandatory or not. To indicate that an extra attribute is mandatory, set the option value to **MANDATORY**. This is an optional entry; if given, the attribute is assumed to be optional. This value may also be set to **EXTERNAL** to indicate that the attribute is not published from this site but should be returned by a search operation. In this case, the **po_pom_attr_name** should be same as **pr_pom_attr_name**.

For example:

form_attr1:security:MANDATORY
immutable:immutable:EXTERNAL

DEFAULT VALUES

None.

If left undefined, no extra attributes are published through preference settings. Publications of extra attributes can still be performed through the **USER_ods_client_publish_extra_attributes** user exit.

DEFAULT PROTECTION SCOPE

Site preference.

TC_ods_client_def_timeout

DESCRIPTION

Sets the basic timeout interval, in seconds, before network nodes (clients) attempting to connect to the Object Directory Services (ODS) terminate the request. This timeout value is used for all subsequent ODS requests.

VALID VALUES

Single numerical value. Siemens Digital Industries Software recommends **300**.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_ods_client_initial_timeout

DESCRIPTION

Sets the basic timeout interval, in seconds, before network nodes (clients) attempting to connect to the Object Directory Services (ODS) terminate the request. This timeout value is used for the initial request to the ODS; this may take longer than subsequent requests.

VALID VALUES

Single numerical value. Siemens Digital Industries Software recommends **300**.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_ods_proxy_server_site_table

DESCRIPTION

Maps the target site ID with its real node name so that the proxy server can redirect a requesting site's first remote procedure call (RPC) message to the target site.

VALID VALUES

This preference must list all sites (both client sites and server sites) that use the proxy host. This list provides the information needed to map site IDs to their real node names. The list is also used to discriminate against clients that are not authorized to use the proxy host. The proxy denies access to any requesting site not on the site table list. For example:

```
TC_ods_proxy_server_site_table=
Site 1 ID:real-node-name-for-Site 1
Site 2 ID:real-node-name-for-Site 2
```

Site ID is the site ID of either an internal or external site that uses the proxy host and *real-node-name-for-Site* is the actual node name for the site ID.

The IP address can be given instead of the node name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_on_demand_sync_broadcast_mode

DESCRIPTION

Controls the report function's query scope when the site known by the current site as the owning site denies ownership.

VALID VALUES

TRUE	Queries all known sites to find the owner (broadcast mode).
FALSE	Performs sequential queries to sites in the ownership chain until it reaches the limit designated by the **TC_follow_ownership_chain_max_site_count** preference.

DEFAULT VALUES

TRUE

DEFAULT PROTECTION SCOPE

Site preference.

TC_ownership_export

DESCRIPTION

Prevents a user from transferring site ownership by disabling the **Transfer Ownership** button in the **Import/Export Options** dialog box.

> Note:
>
> This preference is created when a user selects various import and export commands. It is used internally by Teamcenter and should not be modified.

VALID VALUES

Defined by system.

DEFAULT VALUES

Not applicable.

DEFAULT PROTECTION SCOPE

User preference.

TC_plmxml_import_item_filter

DESCRIPTION

Specifies a property and its associated value used to prevent certain not up-to-date item revisions from being imported during controlled replication. If this preference is not set, all item revisions are imported.

VALID VALUES

Multiple string values delimited by colons (:) representing any **BOMLine** property used as the filtering value in the following format:

```
<identifier>:<value>
```

If multiple values are specified for the same property, the first value processed is used.

DEFAULT VALUES

None

DEFAULT PROTECTION SCOPE

Site preference.

NOTES

For the GM Overlay, the preference is set to **bl_item_object_type:CORP_vehicle**.

TC_plmxml_sync_dataset

DESCRIPTION

Specifies the relation name, dataset type, and the named reference that must be used to create datasets using the existing PLM XML file. Also, relates the PLM XML to the structure context object (SCO), or root item revision of a configured assembly, for which it is generated.

VALID VALUES

String value delimited by colons (:) representing any existing dataset type, relation name, and named reference type in the following format:

```
<dataset-type>:<relation>:<named-reference–type>
```

If a named reference type is not specified, **ConfiguredAssembly** is used.

DEFAULT VALUES

DirectModelAssembly:IMAN_reference:ConfiguredAssembly

DEFAULT PROTECTION SCOPE

Site preference.

RESTRICTIONS

All values in the preference must exist in the Teamcenter database.

NOTES

Modify this preference value when the PLM XML dataset must be attached using a specific relation, dataset type, and named reference for your business needs.

TC_post_export_script

DESCRIPTION

Automatically compresses export data using the **standard_post_export_script.pl** script when a user performs a local export operation using any of the following methods:

Object Export command
item_export utility
Customized ITK program that exports objects via OBJIO functions

If this preference is not defined, no compression occurs.

> Note:
>
> This script does not apply during remote export, which employs a different compression/decompression method.

VALID VALUES

Complete path name of script. If only the script name is given, the script is assumed to be stored in the **TC_BIN** directory.

DEFAULT VALUES

standard_post_export_script.pl (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_pre_import_script

DESCRIPTION

Automatically decompresses export data using the **standard_post_export_script.pl** when a user performs an import operation using any of the following methods:

Files→Import→Objects command
item_export utility
Customized ITK program that exports objects via OBJIO functions

If this preference is not defined, no decompression occurs.

> Note:
>
> This script does not apply during remote import, which employs a different compression/decompression method.

VALID VALUES

Complete path name of script. If only the script name is given, the script is assumed to be stored in the **TC_BIN** directory.

DEFAULT VALUES

standard_post_export_script.pl (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_preserve_original_owner_on_sync

DESCRIPTION

Determines whether the owning user and group of a replica are kept the same as that of the primary copy when synchronizing an object. This preferences is used by the **data_sync** utility to determine the owning user and group at a replica site when synchronizing changes to an object.

VALID VALUES

One of any logical pair (**true** or **false**, **on** or **off**, **0** or **1**).

TRUE The owning user and group are updated at the replica site when the primary object ownership changes, for example:

1. At the owning site, the **obj001** item is owned by the **designer1** user. Replicate the object to the target site and the replica **obj001** object is owned by the **designer1** user.

2. At the owning site, change the **obj001** object owner to the **designer2** user and add a dataset to the item revision also owned by the **designer2** user.

3. Run the **data_sync** utility with the **TC_preserve_original_owner_on_sync** preference set to **TRUE** at the replica site. The owner of the replicated **obj001** item is changed to the **designer2** user and a dataset is added to the item revision also owned by the **designer2** user.

   The owning user and group must be defined at both the owning and replica site to synchronize the ownership at the replica site. Otherwise, the replica ownership does not change.

FALSE The owning user and group are preserved at the replica site when the primary object ownership changes.

DEFAULT VALUES

None.

If no value is defined for this preference, synchronization uses the **FALSE** value behavior.

DEFAULT PROTECTION SCOPE

Site preference.

Tc_preserve_replica_pfmember_ownership

DESCRIPTION

Determines whether part family members created at a site, which are based on a replica part family template, is owned by the owning site or the local site. This preference is applicable when creating part family members from Teamcenter Integration for NX.

VALID VALUES

TRUE	Part family members are created with the same site ownership as that of their template.
FALSE	Part family members are created with the local site ownership.

DEFAULT VALUES

TRUE

DEFAULT PROTECTION SCOPE

Site preference.

TC_publish_item_or_itemrev

DESCRIPTION

Controls the publication behavior of item revision objects. When you select an item revision object to publish, by default, Teamcenter publishes only the item revision object. You can change this behavior to publish the parent item object of the item revision object or both the item revision object and its parent item object.

VALID VALUES

1	Teamcenter publishes only the item revision object.
2	Teamcenter publishes only the parent item of the item revision object.
3	Teamcenter publishes both the parent item and the item revision object.

DEFAULT VALUES

1

DEFAULT PROTECTION SCOPE

Role preference

Tc_publish_pfdata_with_assemblies

DESCRIPTION

Determines how part family data is published when publishing assemblies. This preference can also be set by system administrators in the **Options** dialog box, accessed from the **Edit** menu. General users can set this preference from the **Options** dialog box only if the **Tc_allow_users_edit_pfdata_options** preference is set to **TRUE**.

> Note:
>
> This preference is effective only when publishing assembly level items with the include BOM option.

VALID VALUES

Members	Publishes part family member components in the assembly.
Templates	Publishes part family templates rather than the part family member components in the assembly.
Both	Publishes both part family member components and part family templates in the assembly.
None	Publishes neither part family member components nor part family templates in the assembly.

DEFAULT VALUES

Members

DEFAULT PROTECTION SCOPE

All.

Tc_publish_pfmembers_with_pftemplate

DESCRIPTION

Determines whether to publish related part family member items when publishing part family template items. This preference can also be set by system administrators in the **Options** dialog box, accessed from the **Edit** menu. General users can set this preference from the **Options** dialog box only if the Tc_allow_users_edit_pfdata_options preference is set to **TRUE**.

> Note:
>
> This preference is effective only when publishing part family template items.

VALID VALUES

TRUE	Publishes the corresponding part family member items.
FALSE	Does not publish the corresponding part family member items.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

All.

Tc_publish_pftemplate_with_pfmember

DESCRIPTION

Determines whether to publish corresponding part family templates when publishing part family member items. This preference can also be set by system administrators in the **Options** dialog box, accessed from the **Edit** menu. General users can set this preference from the **Options** dialog box only if the Tc_allow_users_edit_pfdata_options preference is set to **TRUE**.

> Note:
>
> This preference is effective only when publishing part family member items.

VALID VALUES

TRUE	Publishes the corresponding part family templates.
FALSE	Does not publish the corresponding part family templates.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

All.

TC_publishable_classes

DESCRIPTION

Defines Teamcenter classes that can be published to the Object Directory Services (ODS) site.

VALID VALUES

One or more strings. Each string must be one of the following Teamcenter class names or **None**.

Item
Dataset
Form
Folder

If the preference is not defined, only items can be published. If set to a single value of **None**, publication is disabled; other Multi-Site Collaboration operations can still be performed.

DEFAULT VALUES

Item

DEFAULT PROTECTION SCOPE

Site preference.

TC_publishable_<business-object-name>_contexts

DESCRIPTION

Defines contexts to publish for a given business object type. If the context of the identifier object matches the contexts specified in the preference, identifiers that are attached to the business object are published to the Object Directory Service (ODS) when the object is published. You must define this preference to publish alternate ID information to the ODS for a specific type of business object. This allows alternate ID values to be used as search criteria for remote objects.

> Note:
>
> If you set the **ITEM_autopublish** preference to **TRUE**, creating an **Item** object or any **Item** subtype objects publishes the **Item** or **Item** subtype object to the ODS site specified in the **ITEM_autopublish_sites** preference. However, alternate identifiers for the objects are not sent as part of the published information. You must publish the objects manually to publish the alternate ID information.

This preference is not created by the Teamcenter installation process. If you require this functionality, you must create the preference.

For information about creating and using preferences, see *Teamcenter Basics*.

Siemens Digital Industries Software recommends that you use the following values when creating this preference:

- **Protection Scope: Site**
- **Environment: Disabled**
- **Category: MultiSiteCollaboration**

You must select **String** from the **Type** list and **Multiple** from the **Multiple** box.

VALID VALUES

The preference name must be in the form:

`TC_publishable_`*`business-object-name`*`_contexts`

business-object-name represents a valid Teamcenter business object context.

The preference values must be valid Teamcenter identifier types. For example, to publish **Ident0001** identifier types for the **IDContext** business object context:

```
TC_publishable_Ident001_contexts=IDContext001
```

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_reference_update_lmd

DESCRIPTION

Determines whether the last modified date of a locally owned item revision is updated when an object with an associated reference relation is modified.

The last modified date of the parent item is automatically updated as a consequence of the associated item revision update.

VALID VALUES

TRUE	The last modified date of the item revision is modified when an object with an associated reference relation is modified.
FALSE	The last modified date of the item revision is not modified when an object with an associated reference relation is modified.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_rendering_update_lmd

DESCRIPTION

Determines whether the last modified date of a locally owned item revision is updated when an object with an associated rendering relation is modified.

The last modified date of the parent item is automatically updated as a consequence of the associated item revision update.

VALID VALUES

TRUE	The last modified date of the item revision is modified when an object with an associated rendering relation is modified.
FALSE	The last modified date of the item revision is not modified when an object with an associated rendering relation is modified.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_relation_export_on_transfer

DESCRIPTION

Determines which **interpart** relation types are included when an object is exported/imported regardless of whether the export is a replica or with transfer of ownership.

If the exporting site and the importing site have conflicting rules, the rule at the exporting site prevails. For example, if the importing site does not require the manifestation relation but the exporting site does, manifestations are included.

> Caution:
>
> Relations that are in this preference override relations that are also included in the **TC_relation_required_on_transfer** preference.
>
> To avoid having replicated participants missing in the **Assign Participants** dialog box, do not include the **HasParticipant** relation in this preference value.

VALID VALUES

Accepts one or more strings as values. Each string must be a valid Teamcenter relation. An exception is the **IMAN_RES_checkout** relation, which is *not* a valid value for this preference.

DEFAULT VALUES

IMAN_master_form
IMAN_requirement
IMAN_specification
IMAN_ic_intent
IMAN_CCContext
IMAN_StructureContent
IMAN_SCTypeData
TC_Attaches
TCCalendar_Rel_Type

DEFAULT PROTECTION SCOPE

Site preference.

TC_relation_required_on_export

DESCRIPTION

Determines which relation types are included when an object is exported/imported without transfer of site ownership.

If the exporting site and the importing site have conflicting rules, the rule at the exporting site prevails. For example, if the importing site does not require the manifestation relation but the exporting site does, manifestations are included.

> Caution:
>
> To avoid having replicated participants missing in the **Assign Participants** dialog box, do not include the **HasParticipant** relation in this preference value.

VALID VALUES

Accepts one or more strings as values. Each string must be a valid Teamcenter relation. Excepting the **IMAN_RES_audit** and **IMAN_RES_checkout** relations, which are *not* valid values for this preference.

NONE is also a valid value, indicating there are no required relation types for export, except for the implied relation types listed above.

DEFAULT VALUES

IMAN_master_form
IMAN_requirement
IMAN_specification
IMAN_ic_intent
IMAN_CCContext
IMAN_StructureContent
IMAN_SCTypeData
TC_Attaches
TCCalendar_Rel_Type

The **IMAN_master_form** and **IMAN_ic_intent_rtype** relation types are implied entries in this preference and cannot be excluded.

DEFAULT PROTECTION SCOPE

Site preference.

TC_relation_required_on_transfer

DESCRIPTION

Determines which relation types are included when an object is exported/imported with transfer of site ownership.

If the exporting site and the importing site have conflicting rules, the rule at the exporting site prevails. For example, if the importing site does not require the manifestation relation but the exporting site does, manifestations are included.

This preference overrides the **TC_remote_checkin_relation_send_as_replica** preference if the same relation is included in both preferences.

> Caution:
>
> Use this preference for relation types that are required for ownership transfer. Do not include any relation type required for ownership transfer in the **TC_relation_export_on_transfer** preference.

VALID VALUES

Accepts one or more strings as values. Each string must be a valid Teamcenter relation. Excepting the **IMAN_RES_checkout** relation, which is *not* a valid value for this preference.

NONE is also a valid value, indicating there are no required relation types for export, except for the implied relation types listed above.

DEFAULT VALUES

IMAN_master_form
IMAN_requirement
IMAN_specification
IMAN_RES_audit
IMAN_ic_intent
IMAN_CCContext
IMAN_StructureContent
IMAN_SCTypeData
TC_Attaches
TCCalendar_Rel_Type
IMAN_vi_linked_module
IMAN_vi_sos
IMAN_classification
Fnd0ShapeRelation

The **IMAN_master_form**, **IMAN_ic_intent_rtype** and **IMAN_RES** relation types are implied entries in this preference and cannot be excluded.

DEFAULT PROTECTION SCOPE

Site preference.

TC_remote_checkin_assy_option

DESCRIPTION

Determines how the system manages newly added components when a remotely checked out BOM view revision (BVR) or item revision is remotely checked in. The typical use case is for a remote user to remotely check out a BVR to add new components owned by the local site (the site adding the component). When the BVR is remotely checked in, this preference determines how these locally owned components are handled.

VALID VALUES

0	The components are stubbed upon remote check in. A copy is not sent to the site that owns the BVR. Instead, at the owning site, the component is replaced by a POM stub. When the assembly containing the BVR is opened at the owning site via Structure Manager, the application prompts the user to remotely import the component.
1	The components are sent to the owning site as replicas. Site ownership of the components remains with the site performing the remote checkin.
2	Site ownership of the components is transferred to the site that owns the BVR. The components at the site performing the remote check in become replicas.

DEFAULT VALUES

0

DEFAULT PROTECTION SCOPE

User.

TC_remote_checkin_exclude_relations

DESCRIPTION

Determines which relations prompt the system not to transfer attachments as part of the replica when a parent object is remotely checked in.

How the system checks in locally attached objects is determined both by this preference and the relations added to the **Include Reference** list. The **Include Reference** list defines the types of related objects to be imported and exported. The list displays in the **Advanced** tab of the **Multi-Site Collaboration, Import Remote** section of the **Options** dialog box, accessed from the **Edit** menu.

VALID VALUES

Accepts one or more strings as values; each string must be a valid Teamcenter relation.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_remote_checkin_preserve_replica_owning_user_and_group

DESCRIPTION

Determines how ownership is assigned to replica revisions upon checkin. This preference must be set at *both* sites to take effect.

Note:

If this preference is set at only the owning site (and not also at the remote site), the preference acts as if it was not set.

Warning:

Do not set this preference to **TRUE** at the remote site and **FALSE** at the owning site. In this situation, the system displays an error.

VALID VALUES

TRUE The ownership assigned to the replica revision is preserved, regardless of which user checks in the replica.

For example, consider an item created at Site A by User 1. A remote export is performed on the item with transfer of ownership to Site B User 1.

User 1 at Site A then checks out the item, revises **Rev1** of the item, and checks it back in. User 1 at Site A then checks out **Rev2** of the item, changes ownership of the item to User 2 at Site B, and checks it back in. Both Site A and Site B see the ownership of **Rev2** is User 2.

In this example, User 1 must be a valid user at both sites and User 2 must be a valid user at both sites.

FALSE The user checking in the replica is assigned ownership, regardless of any ownership assignments made by the user. This preserves the concept of *unified ownership*.

For example, consider an item created at Site A by User 1. A remote export is performed on the item with transfer of ownership to Site B by User 1.

User 1 at Site A then checks out the item, revises **Rev1** of the item, and checks it back in. User 1 at Site A then checks out **Rev2** of the item, changes ownership of the item to User 2 at Site B, and checks it back in. Site A sees the ownership of **Rev2** as User 2, but Site B sees the ownership of **Rev2** as User 1.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_remote_checkin_relation_send_as_replica

DESCRIPTION

Determines which relations prompt the system to transfer attachments as part of the replica when a parent object is remotely checked in.

How the system checks in locally attached objects is determined both by this preference and the relations added to the **Include Reference** list. The **Include Reference** list defines the types of related objects to be imported and exported. The list displays in the **Advanced** tab of the **Multi-Site Collaboration, Import Remote** section of the **Options** dialog box, accessed from the **Edit** menu.

This preference is overridden by the **TC_relation_required_on_transfer** preference if the same relation is included in both preferences. By default, remote checkin sends locally owned attachments when transferring ownership. This default behavior is overridden if the relation is included in this preference.

VALID VALUES

Accepts one or more strings as values; each string must be a valid Teamcenter relation.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

Site preference.

TC_replication_exclude_types

DESCRIPTION

Specifies a list of item types that are ignored by the replication process that uses PLM XML data contents to individually synchronize objects.

Modify this preference value if you must omit specific item types from replication.

VALID VALUES

Any existing item type.

DEFAULT VALUES

There is no default value.

DEFAULT PROTECTION SCOPE

Site preference.

TC_replica_volume

DESCRIPTION

Identifies the volume where replica files are placed upon import. Use to direct all replica files into a specific volume. This volume does not need to be backed up because they are replica files, you can retrieve a copy from the owning site if you lose any of the files. You can also use operating system facilities to determine which files have not been accessed for a certain period of time, thereby determining unnecessary replicas.

VALID VALUES

Single valid volume name.

DEFAULT VALUES

None.

DEFAULT PROTECTION SCOPE

User preference.

TC_retain_group_on_import

DESCRIPTION

Determines group ownership of the import. When importing an object to a site where the original object's user and group are not defined, the replicated object's owning group is defined as the importing user's group.

To preserve the replica's group ownership even when the original owning user is not defined locally (but the group is defined locally) set to **TRUE**.

VALID VALUES

TRUE	Retains group ownership of the import when the owning group of the original object is define locally.
FALSE	General rules of replica ownership applies.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_split_shared_status_on_replication

DESCRIPTION

Determines whether to split the release status when replicating a object that has a shared release status that references the item revision and its related dataset objects. For exports with ownership transfer, the shared release status is always split.

VALID VALUES

TRUE	Replicated objects have shared release status split from the related objects. The release status references only the item revision at both the remote and owning site.
FALSE	Replicated objects do not have the shared release status object split from the related objects. The release status references the item revision and all related objects at both the remote and owning site.

> Caution:
>
> Setting this preference value to **TRUE** increases the chance of an **instance in use** error occurring during Multi-Site operations, especially when objects with release statuses are replicated at the same time by different users.

DEFAULT VALUES

TRUE

DEFAULT PROTECTION SCOPE

Site preference.

TC_stub_dataset_files_after_ownership_transfer

DESCRIPTION

Specifies whether replica files are stubbed at the original owning site when the ownership is transferred to another site. The Multi-Site transfer of ownership feature creates full **ImanFile** objects at the new owning site and adds full operating system files to the volume. This preference indicates whether the original owning site has the option to stub the replica **ImanFile** objects and purge the files from the volume.

VALID VALUES

TRUE	Allows the replica files to be stubbed at the original owning site.
FALSE	Forces the full object files to remain on the original owning site's volume.

DEFAULT VALUE

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_subscribable_replica_classes

DESCRIPTION

Defines the list of **WorkspaceObject** classes whose replicas can be subscribed to for update notification. When users subscribe to the **Replica Updated** event for a replica object at a replica site, they are notified when the replica is updated due to re-import or synchronization. The notification is sent only for the objects of classes defined by this preference.

VALID VALUES

Accepts multiple strings as values. Each string must be a valid workspace object implementation class, such as **ItemImpl**, **ItemRevisionImpl**, **DatasetImpl**, **DocumentImpl**, **DocumentRevisonImpl**.

> Note:
>
> In the case of the item and item revision classes, Teamcenter accepts the object class (**Item**, **ItemRevision**) in addition to the object implementation class.

DEFAULT VALUES

None. When not defined, no subscription functionality for replicas is enabled.

DEFAULT PROTECTION SCOPE

Site preference.

TC_sync_max_assy_level

DESCRIPTION

Defines the maximum assembly level *reported* when performing synchronization in **Report Only** mode from the **Synchronization Preferences** dialog box. This preference affects only the number of levels displayed in the report, not the maximum assembly level synchronized.

This preference is automatically set by the system when users select the **Save Options As Default** option in the dialog box.

> Tip:
>
> While is possible for users to manually set this preference through the **Options** dialog box accessed from the **Edit** menu, or for administrators to manually edit the **tc_preferences.xml** file, this is not the suggested practice.

The dialog box allows users to perform on-demand synchronization (as opposed to administrative-based synchronization using utilities, or automatic synchronization which only occurs when the primary object is modified.) This functionality provides users with immediate visual confirmation that the synchronization has succeeded or failed. Access the dialog box by choosing **Multi-Site Collaboration→Synchronize** from the **Tools** menu, or right-clicking on object and choosing **Multi-Site Synchronization** from the shortcut menu.

VALID VALUES

-1	All assembly levels are reported.
Any positive integer	The number of levels specified by the integer is reported.

DEFAULT VALUES

-1

DEFAULT PROTECTION SCOPE

User preference.

TC_sync_projects_with_owning_site

DESCRIPTION

Controls the synchronization of the project list of replicas with the project list of the primary object when projects are removed at the owning site.

By default, if one or more projects are removed from the primary object, Teamcenter does not change the project list for the replica object during synchronization. This allows the replica to maintain a project list independent of the primary object. However, if the primary object has no projects assigned, Teamcenter removes all projects from the replica object.

The **TC_sync_projects_with_owning_site** preference overrides the default behavior. For push operations, you must set the preference at the remote IDSM site. For pull operations, you must set the preference at the local importing site.

Set the following values when using the **TC_sync_projects_with_owning_site** preference:

Setting	Value
Protection Scope	**Site** (Recommended)
Category	**Data Sharing.Multi-Site Collaboration** (Recommended)
Environment	**Disabled** (Recommended)
Type	**String**
Multiple	**Multiple**

VALID VALUES

Accepts one or more strings as values. Acceptable values are:

- **FALSE**

  Overrides the default behavior of removing projects from the replica object when there are no projects assigned to the primary object. This allows the replica object to maintain a project list when the primary object has no projects assigned to it.

  FALSE must be the only entry for the preference.

- **ALL**

  Indicates that for all owning sites, when a project is removed from the primary object project list, that project is also removed from the replica object list; and when all projects are removed from the primary object, all projects are removed from the replica object.

- *site-name*

Indicates the **ALL** value behavior applies to the specified site. You can enter a list of sites, one site name per line. Invalid site names are ignored.

The default behavior applies to all sites not specified.

DEFAULT VALUES

None.

If this preference is not set, the default behavior applies to all sites.

DEFAULT PROTECTION SCOPE

Site preference.

TC_sync_revision_rules

DESCRIPTION

Defines the revision rules which display in the **Synchronization Preferences** dialog box. If not defined, all available revision rules appear.

The dialog box allows users to perform on-demand synchronization (as opposed to administrative-based synchronization using utilities, or automatic synchronization which only occurs when the primary object is modified.) This functionality provides users with immediate visual confirmation that the synchronization has succeeded or failed. Access the dialog box by choosing **Multi-Site Collaboration→Synchronize** from the **Tools** menu, or right-clicking an object and choosing **Multi-Site Synchronization** from the shortcut menu.

In the dialog box, users select a revision rule from the **Revision Rule** list. The selected revision rule is passed to the owning site of the selected component and is used by the owning site to determine which item revision to synchronize. It is a required field if the object selected for synchronization is an item.

VALID VALUES

Accepts multiple strings as values. Each string must be a valid Teamcenter revision rule.

DEFAULT VALUES

None. If not defined, all available revision rules appear in the revision rule list.

DEFAULT PROTECTION SCOPE

Site preference.

TC_transfer_area

DESCRIPTION

Sets the directory for temporarily storing data during import and export. If not set, the current working directory is used to store data during import and export.

VALID VALUES

Full OS path to the directory.

This string can be a single line value, for example:

```
TC_transfer_area=
/tmp
```

This string can also contain multiple lines to support a heterogeneous site. In this case, Teamcenter uses the first valid value for the platform (Linux or Windows). Validity is determined by the presence of a backslash (****). For example, where the Windows path uses **C:** as the drive letter, the preference is set for both Linux and Windows as:

```
TC_transfer_area=
/tmp
c:\temp
```

> Note:
>
> UNC paths are specified with a triple backslash. When the Windows path is specified in UNC format, the preference for both Linux and Windows is defined as:
>
> ```
> TC_transfer_area=
> /tmp
> \\\hostx\share_area
> ```

DEFAULT VALUES

/tmp

Typically, the **/tmp** directory is a volatile area of the file system. Files in this directory can be deleted when the system is rebooted. Siemens Digital Industries Software strongly recommends setting up another permanent transfer directory and setting this preference to that directory.

DEFAULT PROTECTION SCOPE

Site preference.

TC_truncate_file_name

DESCRIPTION

Determines if truncation of an original file name is necessary. Implement so Multi-Site Collaboration can transport data between releases.

VALID VALUES

TRUE	Truncates the original file name to the length of 30.
FALSE	Does not truncate the original file name to the length of 30. Only set to **FALSE** if all sites are running V7.0 or higher.

DEFAULT VALUES

TRUE (Commented out).

DEFAULT PROTECTION SCOPE

Site preference.

TC_use_group_admin_as_default_replica_owner

DESCRIPTION

Determines whether the owning user of an imported object is the group administrator of the owning group when the owning user is not a member of the owning group at the importing site.

For example: the primary copy at the owning site (Site 1) is owned by the user Joe, in the Design group, and it is imported at Site 2, where Joe is *not* a member of the Design group. If this preference is set to **TRUE**, the system searches the list of group administrators for the Design group at Site 2, assigning ownership to the first group administrator in the list.

VALID VALUES

TRUE	Assigns ownership of an imported object to the group administrator of the owning group when the owning user is not a member of the owning group at the importing site.
FALSE	General rules of replica ownership applies.

DEFAULT VALUES

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

TC_validate_stub_tickets

DESCRIPTION

Determines whether Teamcenter validates stub tickets and has the site where the associated file was transferred generate a new stub ticket for any found to be invalid.

VALID VALUES

TRUE	Validates stub tickets.
FALSE	Does not validate stub tickets.

DEFAULT VALUE

FALSE

DEFAULT PROTECTION SCOPE

Site preference.

NOTES

The frequency of synchronizing **POM_stub** objects with owning **ImanFile** objects is a determining factor for setting this preference. The overwhelming majority of the time, Teamcenter can find the file in local whole file cache. Rarely, Teamcenter must seek the file in the owning site's volume or in the FSC. There is a chance the file is absent from the owning site's volume due to a recently performed ownership transfer to a third site. To ensure a stub ticket is valid, the Teamcenter site that the file was transferred to must generate the latest stub ticket, which requires a LAN/WAN trip to the owning site. The generated ticket can be used to search the file under the local whole file cache. This sacrifices performance by requiring a LAN/WAN trip to the owning site to generate the ticket for every **POM_stub** object.

Part III: Using Multi-Site Collaboration

Use the information in this section to share objects across your enterprise, import and export objects, share data with unconnected sites, and to leverage other Multi-Site Collaboration features.

Your system administrator must have previously set up and configured Multi-Site Collaboration before you can use these features and functions.

11. Recommended practices for 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.

12. Publishing and unpublishing

Viewing objects that are visible to other sites

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. 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.

Publish and search with multifield keys

Multi-Site requires all participating sites that use an ODS to use the same multifield key definition. When Multi-Site publishes an object that is identified by multifield key attributes, all of the attributes from the publication record are published to the ODS server along with the item ID. When multifield key is enabled, the ODS server can create multiple publication records with the same item ID.

When custom attributes are used as part of the multifield key attributes, the *extensible ODS* feature of Multi-Site must be used to share the attributes among Multi-Site databases. This requires schema configuration and preferences changes at the server and client sites. The publication record schema

must be modified to contain all key attributes. You must add the additional attributes to the **TC_ods_client_extra_attributes** preference that Multi-Site uses in the publication record for an object.

At the ODS site the publication record contains a superset of the attributes at all client sites. At client sites, only the searchable key attributes are required in the publication record. Therefore, you must define the **PublishedObjConfiguredProperties** global constant. Multi-Site uses this constant to map the attributes of a published object created at the client site to the display attributes on the publication record for remote searches.

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.	a. Choose **To Default ODS...** b. Click the **Select Site** button  . c. In the **Site Selection** dialog box, add or remove sites to publish to in the **Selected Sites** list. d. Click **OK**.
Publish to the default ODS with specific selection rules.	a. Choose **To Default ODS...** b. In the **Publish To Default ODS** dialog box, click the **Explore Selected Component(s)** button  . c. In the **Explore** dialog box, select the desired **Selection Rules**. d. Click **OK**.
Publish to all ODS sites in the publication list with the default selection rules.	a. Choose **To ODS Publication List**. b. In the **Publish to Publication List** dialog box, click **Yes**.
Publish to all ODS sites in the publication list with specific selection rules.	a. Choose **To ODS Publication List**.

To	Do this
	b. In the **Publish to Publication List** dialog box, click the **Explore Selected Component(s)** button .
	c. In the **Explore** dialog box, select the desired **Selection Rules**.
	d. 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.	a. Choose **From Default ODS...**
	b. Click the **Select Site** button .
	c. In the **Site Selection** dialog box, add or remove sites to unpublish from in the **Selected Sites** list.
	d. Click **OK**.
	e. Click **Yes**.
Unpublish from all ODS sites without further user actions.	Choose **From All ODS Sites**.
Unpublish from one or more specific ODS sites.	a. Choose **From Specific ODS Site(s)**.
	b. In the **Site Selection** dialog box, add or remove sites to unpublish from in the **Selected Sites** list.
	c. Click **OK**.
	d. 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.

13. Object protection and ownership

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 primary 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 primary 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 primary 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 primary 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** preference is **TRUE** and the owning group is defined at the importing site, original owning group will be preserved.

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 Digital Industries 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.

14. Remote import and export

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
- Owning users and owning groups
- 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
- The default options to use when importing
- Import/export report options
- The related objects to import and export
- Synchronization and notification options
- Which BOMChange objects are included

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 site ownership

Transfer option	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.

Transfer option	General tab
	The **TC_ownership_export** preference controls the default value of this option. Siemens Digital Industries Software recommends that you leave the default setting for this option to **unset**.

Control owning users and owning groups

Send options	Advanced tab
Select New Owning User	Opens the **Organization Selection Dialog** dialog window that allows you to select the owning user from the users available at the exporting site.
Use default user/group ownership rules	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 primary 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 primary 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** preference is defined and set to **TRUE**, and the owning group is defined at the importing site, the original owning group is preserved. 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.

Specify whether remote import or export object operations occur in the background or foreground

Transfer option	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** 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. The **Generate Import/Export Report** option is not supported for background exports. 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: During this operation, the export directory and its contents are given operating system-level access protection based on

Transfer option	General tab
	the protection mask of the **tcserver** process. Exported files are typically accessible to users with Teamcenter administrator 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.

Specify which relationships to include or exclude

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.
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.

Specify which item revision to import or export

Set this option	To do this
Include All Revisions	Export all revisions. When transferring site ownership, this is the only option available.
Latest Revision Only	Export the latest revision regardless of whether it is a working or released revision.
Latest Working Revision Only	Export only the latest working (such as nonreleased) revision.
Latest Working/Any Release Status	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	Export the latest released revision with any release status.
Selected Revision(s) Only	Export only the revision(s) selected in the workspace. This option is not valid for **Remote Import**.
Specific Release Status Only	Export only the latest revision with the given release status selected from the list. This is available only in the rich client.

Specify whether to include assembly components

Structure Manager 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. This option, although similar to the **-include_bom** argument of the **data_share** utility, may not export the same set of objects. The **Include Entire BOM** option traverses all components, subassemblies, and subassembly component relationships. The **-include_bom** argument of the **data_share** utility does not traverse component relationships of related subassemblies. This allows subassemblies and their components to be exported in separate transactions to provide better performance and scalability for very large assemblies.
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.

Structure Manager options	Description
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.
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.

Control import and export runs

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 mutually exclusive with the **Generate Import/Export 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**

Session options	Description
	• **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.
TC XML session options	The transfer option sets available for export. **MultiSiteExpOptSet** Available when using Active Workspace, the rich client, and the command line. When exporting using the command line or the rich client, if an option set is not specified, **MultiSiteExpOptSet** is used. **DefaultMultiSite** Same as **MultiSiteExpOptSet**, except for when chosen in Active Workspace, defined read-only options are unavailable. **MultiSiteGlborgExpSet** Replicates organization data for sites with a **global organization**.

Control synchronization and notification

Multi-Site Collaboration **Synchronization/Notification** options make it possible to control how the replica is synchronized when the primary 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 primary 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 primary copy is modified.

Specify which dataset versions and named references to include

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. 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.

Specify the default options to use when importing

Save options	Description
Save All Options As Default	Saves the selected options as the default behavior when importing remote objects.

Control 4GD and design element export

The **Custom** tab shown for remote export, remote import, and remote check on the **Import Export Options** dialog box is dynamically populated. The content is determined by user selections made in the

TC XML session options list on the **General** tab. The options are listed under the appropriate group on the dialog box. The following are standard options in the 4GD group.

4GD options (on Custom tab)	Description
Export realized BOM with Design Element	By default, Multi-Site exports BOM structure separately from the 4GD structure. Select this option If you want to export BOM structure and 4GD structure together in single operation.
Export realized Design Element with Workset/ subset	By default, Multi-Site exports workset and subset structure separately from design elements. Select this option to export workset and subset structure and design elements together in a single operation.

Specify which related objects are imported and exported

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.

Specify which BOMChange objects are included

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.
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

Change objects options	Description
	supersedure information can only be transferred within the context of a **BOMChange**. To include a form associated with the supersedure, the **Cm0BOMHasSupersedureForm** relation must be added to the **Include Reference** list.

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 Digital Industries Software recommends that database sites use rules-based object protection to ensure that released

Teamcenter object	Import export behavior
	objects are protected. Otherwise, when using object-based protection, the released objects inherit the default ACLs of the person performing the remote import.
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 BOM view 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.`

Teamcenter object	Import export behavior
	When evaluating variant conditions, remote/unreadable options are interpreted as undefined (a ? appears in the **Is Configured** column, regardless of the rule).
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** check box, 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. You can export custom notes on trace links to Microsoft Word. Teamcenter supplies the **REQ_export_notesonlinks** transfer mode for exporting custom notes. You cannot import custom notes using PLM XML.

15. Remote checkin and checkout

When to use remote checkin/checkout over transfer of site ownership

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.

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 *Modifying remote objects*.
- The change to be made does not include making variant changes against a replica. (The inability to make variant changes against a replica is a limitation of remote CICO.)
- 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.

> Note:
>
> Revising a remote checkout object does not support any of the **Advanced** tab features that are available when you revise a locally owned object.

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 primary copy at the owning site. The reservation not only prevents other users from checking out the primary copy, but also from transferring site ownership. This effectively puts

a lock on the primary 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 primary 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.

Objects that can be checked out remotely

Remote checkout of objects is limited to the following classes:

- **BOM view**
- **BOM view revision**
- **Dataset**
- **Folder**
- **Form**
- **Item**
- **Item revision**

Classes that can be added to checked out objects

In general, only generic classes such as items, revisions, datasets, and forms can be added to a checked out object. The following classes cannot be added to a checked out object and later checked in:

- **Engineering change**
- **Incremental change**
- **Variants**
- **Absolute occurrences**
- **Teamcenter Mechatronics Process Management classes**
- **Manufacturing Process Management classes**
- **Classification data**

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

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, the changed item revision is imported to the owning site, then checked in to the owning site, and finally 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.

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 or remote export of an NX assembly that contains arrangements:

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

Select these relations on the **Advanced** tab of the **Remote Export Options** dialog.

> Note:
>
> There may be undesired behavior due to including these relationships in the required on export or required on transfer preferences.

Remote CICO and the 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:

Use this option	To do this
list_remote_co	At the owning site, list the objects that are checked out by remote users.
list_replica_co	At the replica site, list the replica objects that are checked out from their respective owning sites.
cancel_remote_co	At the owning site, 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 primary 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	At the replica site, cancel the checkout on a replica. Use this only if it is known that the replica is checked out but the primary copy is not checked out. To cancel a checkout where both the replica and primary copy are checked out, use the **Tools→Checkin/Checkout→Cancel Checkout** menu option.

The **data_share** utility supports 4th Generation Design data using a TC XML payload and the island of data concept to checkout and checkin the 4GD object (primary) and its dependent (secondary) objects.

The **data_share** utility help (**-h** argument) provides additional information.

16. Importing remote objects

Finding objects to import

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.

Searching for remote items

Use the following steps to search for remote objects using **Remote** search.

1. On the rich client toolbar, click **Open Search View** .
2. In the **Search** pane, click and select **Remote**. The remote search criteria are displayed.
3. Enter or select the search criteria for your remote object and click to perform the search.

Import-related 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.

If you use the **Save all options as default** option during a rich client transfer, the **TC_relation_export** preference is automatically defined at your site. This is a system-generated preference that must not be modified by an administrator or other user. When this preference is defined, it also affects transfers you perform using the **data_share** utility. All relations except the ones defined in this preference are excluded from the replication or transfer.

However, any relations specified in the **TC_relation_required_on_export** and **TC_relation_required_on_transfer** preferences are included as these override the **TC_relation_export** preference values. You can also include or exclude specific relations using the **-include** and **-exclude** arguments of the **data_share** utility, which override the values in the **TC_relation_export** preference, but do not override the other preferences' values.

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** tab 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.

If your users replicate NX assemblies with arrangements or use remote checkin or checkout on such assemblies, include the following relations in transfers:

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

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, 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 primary copy becomes a read-only replica. The object in your database becomes the primary 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`
Released objects	Released objects are transferable. The release status and audit file are exported. Siemens Digital Industries 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.

Object	Behavior description

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).

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 are not 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 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 Digital Industries Software recommends that projects be duplicated across sites before attempting to share project data. Import of project assignment data at a remote site is performed as the IDSM user. Therefore, either the IDSM user must be added to the project at the remote site, or the

Object	Behavior description
	TC_multi_site_project_member_bypass preference must be set to **TRUE** for the project ID assignment to appear at the remote site.
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** check box, 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, by 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. With the remote object selected, choose **Tools→Import→Remote**. The **Import Remote** dialog box is displayed.

2. Enter a reason for importing the remote object.

3. Optionally click **Import Remote Option Settings**  to set any remote import options.

4. Click **Yes** to start the remote import operation. The **Import Remote Options Setting** dialog box is displayed with the current remote import option settings.

5. Click **Yes** to continue with the remote import. The **Remote Import Progress** dialog box shows the object name, operation, and progress status of both active and completed remote import operations.

17. Modifying remote objects

Modifying data currently owned by another site

Multi-Site Collaboration provides two methods for granting 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.

Modify attachments on remote objects

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

The following use case illustrates modifying a specification dataset of a remote item revision:

Step	Remote site	Owning site
1	Replicate the item, item revision, and the desired attachment.	Requested objects are exported and sent to requesting site.
2	Check out the specification dataset. Because the dataset is a replica, the	The IDSM checks out the dataset on behalf of the remote user. At this point, no other user

Step	Remote site	Owning site
	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.	(either remote or local to the owning site) can check out the dataset. The item site ownership cannot be transferred.
3	If you decide 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	Modify the dataset.	
5	Check 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.

Add a new item revision

The IDSM user must have **Write** access to a primary 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, 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.

The following use case illustrates adding a new item revision to an existing remote item:

Step	Remote site	Owning site
1	Replicate the item and at least one item revision.	Requested objects are exported and sent to requesting site.
2	Perform a remote check out on the item and then performs a revise action on the item revision. 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	The IDSM exports the new item revision and sends it to the remote site.

Step	Remote site	Owning site
	request for the new item revision is automatically issued.	
2b	Upon successful import of the new item revision, the window is refreshed to show the new item revision.	
3	You can perform a remote checkout of the item revision and make the modifications on the newly created item revision.	
Notes:	If you click **Assign** 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.	You must implement a means to integrate custom part numbering schemes.

Add components to an item with no existing BOM view

In the following use case, you add components to an existing item revision owned by another site. The item revision does not contain a BOM view. Perform the checkout from My Teamcenter.

Step	Remote site	Owning site
1	Replicate the item and the item revision to which components will be added. Expand the item and verify no BOM view exists.	Requested objects are exported and sent to requesting site.
2	Check 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	Select 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,	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.

Step	Remote site	Owning site
	thereby ensuring that no other user at this site can check out the replica item revision.	
4	Send the item revision to Structure Manager and add the necessary components. Structure Manager displays a message stating that no structure data exists and provides an option to create them.	
5	After all components are added, exit Structure Manager and check 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	Check 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, you add components to an existing item revision owned by another site. The item revision contains a BOM view, but no BOM view revision. Performed the check out from My Teamcenter:

Step	Remote site	Owning site
1	Replicate the item and the item revision where you want to add components. Expand the item and verify that a BOM view exists. Expand the item revision and verify that no BOM view revision exists.	Requested objects are exported and sent to requesting site.
2	Check 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.

Step	Remote site	Owning site
3	Send the item revision to Structure Manager and add the necessary components. Structure Manager displays a message stating that no structure data exists and provides an option to create them.	
4	After all components are added, exit Structure Manager and check 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, you add components to an existing item revision owned by another site. The item revision contains a BOM view revision. Perform the check out from My Teamcenter:

Step	Remote site	Owning site
1	Replicate the item and the item revision where you want to add components. Expand the item and verify that a BOM view exists. Expand the item revision and verify that a BOM view revision exists.	Requested objects are exported and sent to requesting site.
2	Check 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	Check 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.

Step	Remote site	Owning site
4	Sends the item revision to Structure Manager and add the necessary components.	
5	After adding all components, exit Structure Manager and check 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	Check in the item.	The IDSM checks in the item.

Add components using Teamcenter Integration for NX

The following example adds a component to a remote item revision using NX. Before running NX, the you 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.

Teamcenter Integration for NX always attempts an implicit checkout using a special reservation type that does not perform a remote checkout. Therefore, you 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 does not occur using the implicit checkin that Teamcenter Integration for NX routinely performs.

Note:

Be aware that a new version of a remote object's dataset is created each time the remote object is checked out. The maximum number of dataset versions maintained on the remote site is controlled by the **AE_dataset_default_keep_limit** preference. (The default value is **10**.) When the number of versions exceeds this value, the oldest version is purged and cannot be recovered. Ensure the value of **AE_dataset_default_keep_limit** is set higher than the number of versions you may need to restore.

Step	Remote site	Owning site
1	Replicate the item, item revision and the **UGMASTER** dataset of the desired item revision.	Requested objects are exported and sent to requesting site.
2	Check 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.
3	Check 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, retrieve 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 to you and, if so, allows NX to modify the replica.	
5	Add components to the assembly. You may save the changes and exit NX, then continue with additional changes at a later time. No remote check in is performed during the save operations.	
6	After all changes are made, check 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.

Baseline automatic remote checkin/checkout 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 WIP designs with suppliers/OEMs.
- Capture alternative designs created during product development.
- Preserve WIP data for historical purposes.

Baselines do not contain the attachments that are excluded during remote import at the replica site.

Baselines are an image of the item revision as seen at the remote site and not at the owning site. If there are attachments that are excluded during the remote import at the replica site, then the baseline does not contain these attachments, irrespective of the business rules that are specified.

A 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:

- When the structure has components from many sites, the baseline must be created only at the owning site, due to excessive network traffic.
- When rollback of a 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 **Baseline_auto_remote_checkout_allowed** preference allows you to perform automatic 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 a piece part automatic remote check out, 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.

When the baseline operation is complete, it must check in the objects that were explicitly checked out.

If you encounter errors during the baseline creation, you may rollback using **Remote Cancel Check Out** on the objects that were **Remote Checked Out**.

Delete replicas to allow deleting a primary object

Before you can delete a primary object from a Multi-Site environment, you must delete all replicas within the environment.

1. Ensure there are no dependencies at the remote site by deleting the item, any datasets, and any item revisions that exist for the replica.

2. At the owning site, use the **data_sync** utility to update the item with the **-update** and **-verify** arguments:

   ```
   data_sync -u=tc-admin-user -p=password -g=group -verify -update
   -site=owning-site -item_id=item-id
   ```

When you use the **-verify** argument with the **data_sync** utility, the utility checks for the existence of the exported object at remote sites. If it does not find a replica at any remote site, Teamcenter deletes the import export record (IXR) for the replica at the owning site.

You can now delete the primary object at the owning site.

18. Sharing data with unconnected sites

Methods for sharing data with offline sites

If your Multi-Site Collaboration network includes sites that are not connected through LAN or WAN (offline sites), you can 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.

You can also export and import data to offline sites using a briefcase file. If your environment has Teamcenter Integration Framework installed, ensure that the **GMS_offline_use_TcGS** preference is set to **FALSE**.

Following are the differences between a Teamcenter Integration Framework briefcase transfer and a Multi-Site (non-Teamcenter Integration Framework) briefcase transfer.

Feature	Teamcenter Integration Framework transfer	Multi-Site transfer
Transaction	Asynchronous	Synchronous
Transaction history	Maintained in the Teamcenter Integration Framework datastore (database) and can be access through the Teamcenter Integration Framework activity status page.	None
Briefcase location	Created and shown in Teamcenter.	Created and shown in Teamcenter with the option to save in the operating system file structure (folder or directory).
Mapping	A separate step in the Teamcenter Integration Framework process that is configurable.	Requires a pre/post action in import/export by attaching an XSL transform to the transfer mode.
Configuration	Teamcenter and Teamcenter Integration Framework required.	Two-tier or four-tier Teamcenter required.
Log access	Accessible through Teamcenter Integration Framework activity page link to log files.	Export and import operations display a dialog box at the end of the execution for viewing and accessing the log files.

Briefcase transfers

Transferring a briefcase package (sites not using Teamcenter Integration Framework)

The process of transferring data for reference using a briefcase package varies depending on whether your environment has a hub site and whether it is transferred for reference only or for modification.

Teamcenter recognizes files with **bcz** and **ugs** extensions as briefcase files.

Transfer process through a hub site

1. Package the data in a briefcase file with the hub site as the target site.
2. Transfer the file to the hub site using FTP or a physical process.
3. Import the file at the hub site.
4. At the hub site, package the data in a briefcase file with the destination (supplier) site as the target.
5. Transfer the file to the destination site using FTP or a physical process.
6. Import the file at the destination site.
7. After reference or modification, package the data in a briefcase file with the hub site as the target site.
8. Transfer the file to the hub site using FTP or a physical process.
9. At the hub site package the data in a briefcase file with the originating site as the target site.
10. Transfer the file to the originating site using FTP or a physical process.
11. Import the file at the originating site.

Transfer process directly to an offline site

1. Package the data in a briefcase file with the destination (suppler) site as the target site.
2. Transfer the file to the supplier site using FTP or a physical process.
3. Import the file at the supplier site.
4. After reference or modification, package the data in a briefcase file with the originating site as the target site.

5. Transfer the file to the originating site using FTP or a physical process.

6. Import the file at the originating site.

Package the data

1. If the transfer is for modification, perform the following steps for each object to be modified:

   > Caution:
   >
   > If you are transferring through a hub site, perform this step only at the hub site. Do not check out objects to a site from the originating site.

   a. In My Teamcenter, select the object and choose **Tools→Site Check-In/Out→Check-out To Site**. The **Check-out To Site** dialog box is displayed.

   b. From the **Target Site** list, select the desired site. Optionally change the ID and enter comment text.

   c. Click **Yes** to check out the object to the site.

2. Select the object to be packaged in the briefcase file.

3. Choose **Tools→Export→To Briefcase**.

4. In the **Export to Briefcase** dialog box, if the desired destination or hub site is not listed in **Target Sites**, click  to display the **Remote Site Selection** dialog box. Select the desired site from the **Available Sites** list or select **Any Site** to create a standard package file and click **OK**.

5. Select the **TIEUnconfiguredExportDefault** transfer option set.

   For exporting a configured bill of materials, see *Exporting a configured bill of materials (BOM)* in *Teamcenter Data Exchange*.

6. (Optional) Click Display/set import options  to set specific options for the export. The options available depend on the selected transfer option set.

7. Set **Export Directory** and **Export Filename** as desired.

8. Click **OK** and confirm your export settings. The export starts and the **Export to Briefcase** dialog box remains open until the export completes.

Import the package file

1. In My Teamcenter, choose **Tools→Import→From Briefcase**. The **Import Briefcase** dialog box is displayed.

2. Specify the Briefcase package file in **Briefcase File**.

3. Select **TIEImportOptionSetDefault** from the **Option Set** list.

4. (Optional) If the Briefcase file contains parts that were exported from this site with objects that were checked out to a site, select the **Site Check-In after import** check box.

   In the **TIEImportOptionSetDefault** dialog box, select the desired import options.

5. Click **OK**, confirm your options, and import the briefcase package file.

19. Updating an object or BOM

Updating an object or BOM

Updating remote objects ensures that the replicated objects in your local database are completely current.

You can optionally choose to retrieve additional information (for example, the entire BOM and all dataset versions), but be aware this increases the amount of local data you store 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.

Update a remote BOM

To view a 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 view opens, it is complete and you do not need to update the remote BOM with the steps in this procedure.

   - If this BOM is incomplete, the following message is displayed:

     ```
     The BOM view contains remote items
     Do you want to import them?
     ```

2. Click **Yes**. The **Objects** dialog box lists the 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**.

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**.

3. If you want to use non-default import options, click **Set remote import options** and set your import options. Unless you have site-specific reasons to use different settings, Siemens Digital Industries Software recommends that you use the following option settings:

Transfer Ownership	Unset
Include Entire BOM	Unset
Include All Versions	Unset
Include All Files	Set

   Click **OK** to save the option settings.

4. In the **Import from PLM** dialog box, click **Yes**.

20. Using synchronization

Check replica synchronization

The replica sync state of an object identifies whether it is up to date with the primary object.

1. Select the replica objects that you want to check.

2. Choose **Tools→Multi-Site Collaboration→Check Replica Sync State**.

   Teamcenter displays the **Replica Sync State** dialog box showing replicated object status.

Synchronization options

You set synchronization options in the **Import Remote Options** dialog box. You can choose between *automatic* synchronization and *batch* synchronization.

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

Additionally, you can request to be notified when the primary 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 primary 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 primary object is modified. This option can be used at any site. If the **Notify By**

Option	Results
	E-mail option is also defined, you are notified when the primary object is modified. This is a read-only option that is controlled by the TC_sync_auto_synchronize preference.
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 primary object is modified.
Notify By E-mail	You are notified by system e-mail when the primary 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** and TC_subscribable_replica_classes and preferences.

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.

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.

Synchronize visualization data only

You use the **-OnlyVIS** argument with the **data_sync** utility to synchronize only the visualization datasets that are 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**). Doing so causes synchronization of visualization data that has been modified after the date. For example:

```
data_sync -u=tc-admin-user -p=password -g=group -OnlyVIS -include=IMAN_Rendering
-include=IMAN_Manifestation -since=2021-05-01:01:01
-site=cologneIdsm -sync -update -report=report.lst
```

Synchronize bulk data only

Use the **data_sync** utility to synchronize bulk data copied to **Site1** and output a report to the **report.lst** file, for example:

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

The following example synchronizes bulk data with the datasets names listed in the file **ListOfDataSets.txt**:

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

Synchronizing objects on-demand

The rich client provides **Tools** menu commands that allow you to obtain the synchronization state or to synchronize a specific component or assembly. Synchronization state indicates whether the replica object is up to date or whether an object that has been added to the primary object 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.
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** 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.

Select this	To
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 owned by sites other than the site from which you are importing an assembly are included. This includes distributed components within a distributed assembly (an assembly in which components are owned by more than one site). During synchronization, the top-level assembly and all components owned by the assembly owning site are retrieved first. 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**.
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**.
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. Right-click on the component in My Teamcenter and choose **Multi-Site Synchronization→Object**.
2. Click Synchronization preferences and select **Perform Sync** or **Perform Sync in Background**.
3. If the component selected is not an item revision, select **Specific Revision Rule** and select a revision rule from the list.
4. Select the other options you desire and click **OK**.
5. 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. 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 *Synchronizing objects on-demand*.

From Structure Manager, you can use on-demand synchronization with the **Report Only** option to verify synchronization of the assembly and its components.

You can also use the **sync_on_demand** utility to perform these functions from the command line.

Automatic synchronization

The user that replicates an object can specify that the replica be synchronized automatically when the primary 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 primary 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 primary copy is modified. This option requires the Subscription Manager at both the owning and replica site.
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 primary 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** preference to **ON**. For notification to occur, subscriptions must also be enabled at the importing site.

21. Archiving and restoring data

Overview of archiving and restoring data using Multi-Site Collaboration

Multi-Site Collaboration allows you to archive older, infrequently-used data that can consume significant database table space and file system resources, causing performance degradation when searches traverse this data. Multi-Site Collaboration archiving:

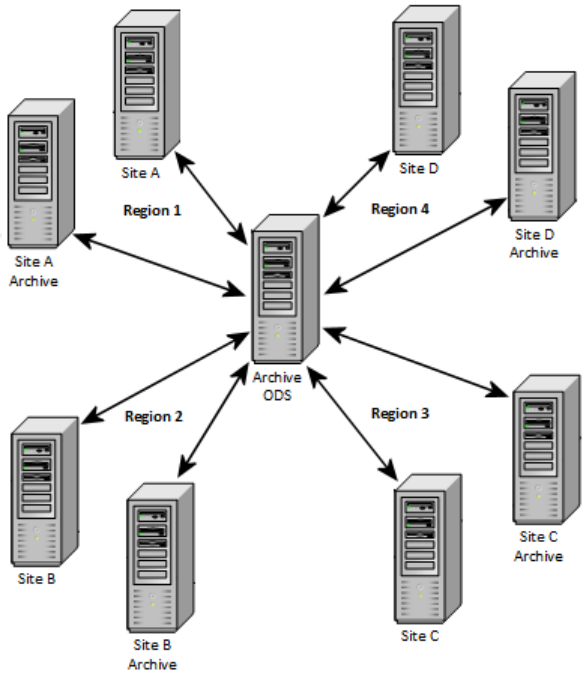
- Improves system usability as users do not need to sift through inactive data to find what they need.
- Improves system performance as results are retrieved more quickly through optimized database indexes.
- Meets technical, legal, and other business requirements for long-term data storage.
- Provides easy access to archived data for recovery, reuse, or repurposing.

Database objects from particular owning sites in a Multi-Site federation are archived to mirror archive sites in the federation. Only an owning site's local objects can be archived to a single archive site. Objects can be restored only to the site from which they were archived. Object Directory Services (ODS) are used to ensure uniqueness of objects that have been archived.

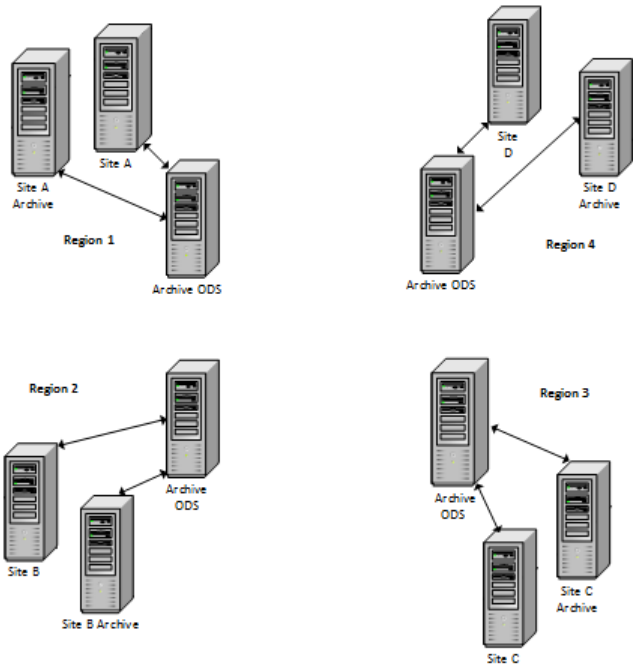
Archiving deployment options are very flexible. Approaches are typically of two types:

Multiple archive sites

In this scenario, each site has a dedicated archive site. Organizations that have relatively infrequent archiving needs could choose to share a single archive ODS between multiple sites as shown in the following layout.

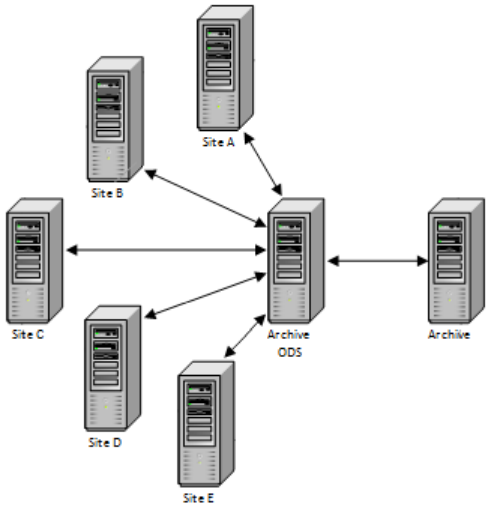


Organizations with more frequent archiving needs that could benefit from dedicated archive ODS sites, or who have need to restrict the location of archived data could adopt the following layout.



Shared single archive site

Organizations desiring a centralized archive could share a single archive site as shown in the following layout.



Archiving does not duplicate the data. Objects are copied as replicas to the archive site and then deleted from the owning site. Restoring also does not duplicate the data. Objects are restored to the owning site and deleted from the archive site.

Teamcenter objects belonging to either a class or a subclass of **Item** or **Mdl0ApplicationModel** can be archived.

You can archive and restore objects from the Teamcenter rich client and from the command line. You can also archive objects using a workflow handler. When objects are archived, they are published for future search and restore operations. When restoring objects that were published when archived, the archived objects are unpublished.

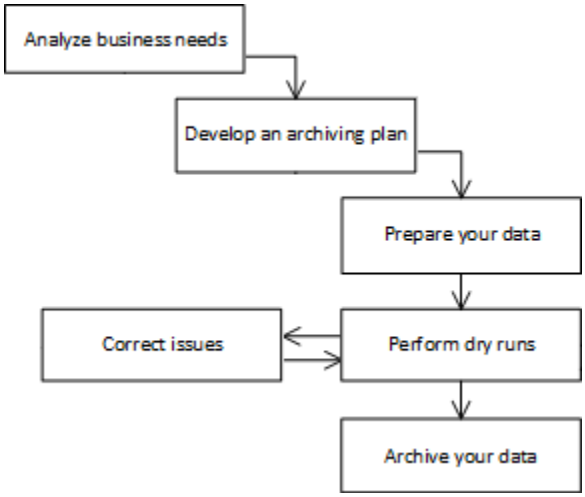
Archiving revisions

In addition to items, you can archive item revisions up to the latest revision. Archiving the latest revision also archives the item.

If the latest revision is archived, you cannot restore an earlier revision.

Archiving process

Once **archiving and restoring with Multi-Site Collaboration is enabled at your site**, the archiving process involves more than the act of archiving data. Planning for archiving, preparing your data, and other steps are involved, as shown in the following process flow.



Consider the following items when employing this process:

Analyze your business needs

Data archiving is designed for data that does not have current or planned specific future needs. Archiving completely removes data from the owning site that is not expected to need to be retrieved. Archiving is not a means to back up data that is being used or has future business needs.

Identify the data at your site that meets these criteria. Often this identification occurs as part of a broader corporate plan regarding data retention. Ensure your plan fits within any broader plans at your organization.

When identifying data, consider the following items:

- Data that has not been modified in several months or years is a better candidate for archiving than data recently modified.
- Data for products no longer supported is a better candidate than data for products still under support.
- Regulatory compliance laws may dictate data required to be archived.

For the data targeted for archiving, assign a retention schedule. Some data may need to be archived indefinitely. Other data may need to be removed from an archive to meet regulatory requirements after a certain period of time.

Develop an archiving plan

Several considerations go into planning archiving. Networking infrastructures, data volume, archiving frequency, office locations, and other items all factor into an archiving plan. Following is a partial list of items to consider when creating your archiving plan.

- Consider the number and location of archive servers. Some corporations may benefit from a single centralized archive. Others may benefit from several regional archive servers, each serving as an archive for several sites. Other sites may benefit from an archive server at every site. Weigh

the benefits of network traffic, data location, data volume, and your site's existing infrastructure when determining the number and location of archive servers.

- Establish expectations and a schedule for archiving. Determine how often archiving will be necessary, and whether manually archiving data on a periodic basis is sufficient, or whether archiving should be set up as an automated task that occurs on a regular schedule.

Prepare your data

Before archiving your data, ensure it is ready to be moved to the archive server.

- Archived data must be unique. Remove unnecessary data by deleting any replicas in the Multi-Site federation of data that will be archived. See *Replica deletion process* for details on deleting replicas and using the **delete_replica** utility.

- Consider unpublishing any of the identified data that has been published and made available to other sites. See *Unpublish an object* for details on unpublishing objects. (If you do not manually unpublish the data, the archiving process will automatically unpublish the data and convert any references to POM stubs.)

Perform dry runs

Use the archiving dry run feature to ensure the archive results will be what you expect them to be. Make any necessary changes to your data and configuration until the dry runs return the results you expect.

Archive your data

Archive your data using the steps detailed in *Archive data using Multi-Site Collaboration*

Archive data using Multi-Site Collaboration

Archiving data is one part of a broader archiving process that involves archive planning and data preparation. Ensure you have performed these steps as described in **Archiving process** before archiving your data.

Use the following steps to archive data with the Teamcenter rich client. You can also archive data from the command line using the utilities described in **Archive and restore data from the command line**. Siemens Digital Industries Software recommends you perform archiving dry runs as described in the following procedure to check for any unexpected results before performing an actual archiving run.

1. Log on to Teamcenter on the owning site as a user with administrative and Access Manager bypass privileges.

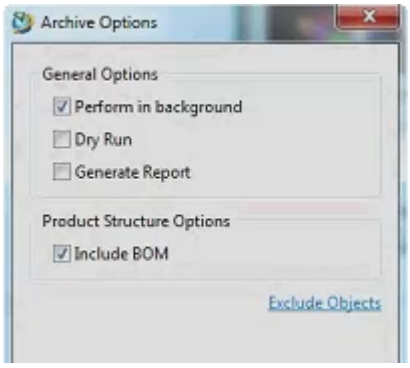
2. Navigate to and select the assemblies and objects, or their revisions, that you wish to archive from the owning site.

   Items or item revisions to archive must be either a class or subclass of **Item** or **Mdl0ApplicationModel**. Archiving the last remaining revision of an object archives the object.

3. Select **Tools** > **Multi-Site Collaboration** > **Unused Data** > **Archive**.

   The **Archive** menu choice is only available when the selected objects are either a class or subclass of **Item** or **Mdl0ApplicationModel**.

   Teamcenter displays the **Archive Options** dialog box.

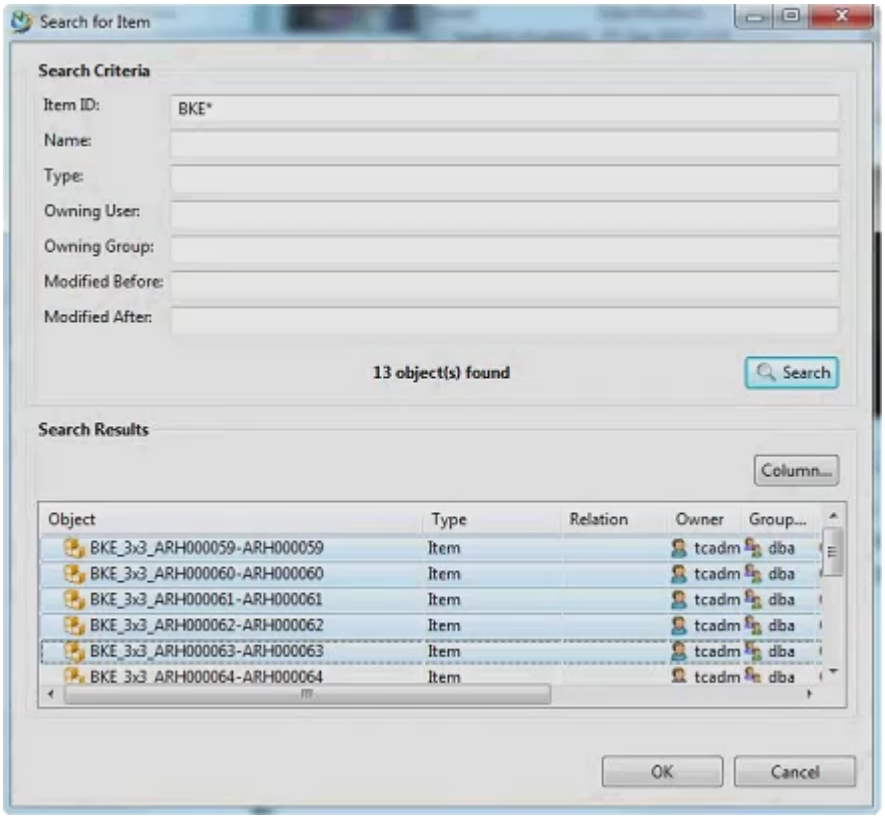


4. In the **Archive Options** dialog box, select the options that you want to apply to the archiving run.

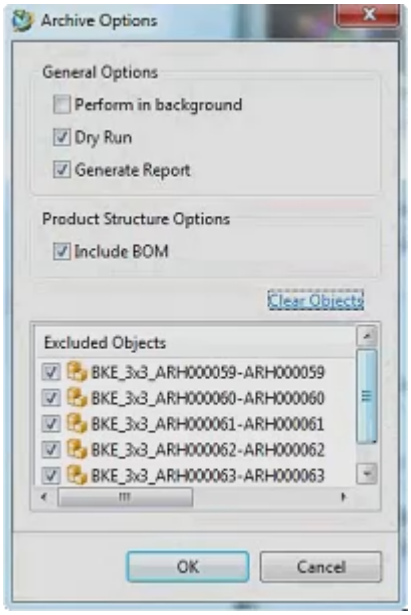
Select this option	To do this
Perform in background	Archive as a background task, freeing Teamcenter for use while the archiving is occurring.
Dry Run	Archive in a test mode. The connection to the archive site is verified and the data is checked for potential error conditions without objects actually being moved to the archive site. When selected with **Generate Report**, **Dry Run** lets you verify that the archive results will be what you expect them to be.
Generate Report	At the end of the archiving run, generate a list showing the archived objects and related archiving information.
Include BOM	Archive BOM assembly components at all levels of the structure. This option does not apply to revisions or 4GD objects.

5. If you want to exclude a subset of selected objects from archiving, then click **Exclude Objects**.

   In the **Search for Item** dialog box, search for the items you wish to exclude. From the **Search Results** list, select the specific objects to exclude.



After selecting the objects to exclude, click **OK** to redisplay the **Archive Options** dialog box. The objects to exclude are listed.



Uncheck any objects that you no longer want to exclude.

To remove all the objects from the **Excluded Objects** list, click **Clear Objects**.

6. Click **OK** to perform the archive.

   Once the objects are archived on the archive site, ownership is transferred to the archive site and the objects are removed from the owning site. Any production data references to archived data are converted to POM stub references.

7. Review the data on the owning and archive sites to confirm the archiving results.

Restore data using Multi-Site Collaboration

When restoring data, restore to the original site from which the data was archived. In rare cases where the original site no longer exists, refer to *Restore to an alternative site using Multi-Site Collaboration* for instructions on restoring the data.

Use the following steps to restore data with the Teamcenter rich client. You can also restore data from the command line with the utilities described in *Archive and restore data from the command line*. Siemens Digital Industries Software recommends you perform dry runs of the restore operation as described in the following procedure to check for any unexpected results before performing an actual restore run.

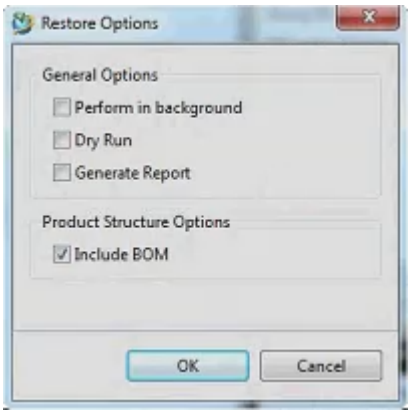
1. Log on to the owning Teamcenter site as a user with administrative and Access Manager bypass privileges.

2. Locate the objects to restore by **searching remote items** owned by the archive site connected to this owning site.

   Select the assemblies and objects that you wish to restore.

3. Select **Tools** > **Multi-Site Collaboration**> **Unused Data** > **Restore**.

   Teamcenter displays the **Restore Options** dialog box.



4. On the **Restore Options** dialog box, select the options you want to apply when restoring.

Select this option	To do this
Perform in background	Restore as a background task, freeing Teamcenter for use while the restoring is occurring.
Dry Run	Restore in a test mode. The connection between the owning site and the archive site is verified and the data is checked for potential error conditions without objects actually being restored to the owning site. When selected with **Generate Report**, **Dry Run** lets you verify that the restore results will be what you expect them to be.
Generate Report	At the end of the restoring run, generate a list showing the restored objects and related restore information.
Include BOM	Restore BOM assembly components at all levels of the structure. This option does not apply to revisions or 4GD objects.

5. Click **OK** to perform the restore. Once the objects are restored on the owning sites, they are removed from the archive site.

6. Search for the restored objects on the owning site and integrate them in your projects as needed.

Restore to an alternative site using Multi-Site Collaboration

In rare cases, data that was archived from a site that has since been retired or otherwise become unavailable may need to be restored. Use the following procedure to restore the archived data to an alternative site.

1. Identify an existing site in or add a new site to the Multi-Site Collaboration federation to act as the replacement site for the retired site.

2. Use the **ar_recover** utility's **-update**, **-extinct_site**, and **-new_site** arguments to allow restoring to the replacement site.

   For example, if site A is retired and site B is identified as the replacement site, the following command would configure site B to allow restoring:

   ```
   ar_recover -f=update -extinct_siteID=siteA -new_siteID=siteB
   ```

   Refer to **ar_recover** for details on the **ar_recover** utility.

Archive and restore data from the command line

Multi-Site Collaboration archive and restore capabilities are available from the command line through the following related utilities.

- **archive**

  Prepares, transfers, and publishes the input objects and their dependents to the archive site with ownership privileges. If required, use the **TC_AR_Excluded_Objs_Folder** preference to designate a unique folder containing objects to be excluded from the archiving.

- **restore**

  Restores the objects to the current site along with their dependent objects with ownership privileges.

- **ar_recover**

  Performs a recovery and cleanup of partially-created data during Multi-Site Collaboration archive and restore operations when IDSM an ODS connection failures have occurred.

- **site_util**

  The **-archive** argument specifies whether the site being created or modified is a Multi-Site Collaboration archive site.

- **delete_replica**

  The **-4gd_id**, **-class**, and **-classoffile** arguments are used to specify replica 4GD objects for deletion.

Archive using workflow handlers

Use the workflow handler **OBJIO-archive-target-objects** to non-interactively perform archiving of objects as part of work processes. The **OBJIO-archive-target-objects** user executing the handler must be a system administrator with DBA privileges. (The user cannot be **infodba**.)

Maintain archive sites

Once you have implemented Multi-Site Collaboration archiving, employ the following practices to maintain your sites:

- When you add features to your production database, also add those features to the archive database.
- Keep your archive and production data models synchronized.
- When new users and groups are added to the production site, also add them to the archive site.
- Keep production sites and archive sites at the same software release levels. When you patch a production site, also patch the related archive site.

  When you upgrade, upgrade your production database before your archive database. Then, reestablish the connection between the sites using the TEM Maintenance option.

- Keep access to your archive sites isolated and restricted to users with dba group privileges. Take care to protect your archived data from accidental deletion or corruption.

- If a site is designated as an archive site, it should be exclusively used for archive and restore operations.

22. Using a remote inbox

If you have the appropriate privileges, you can subscribe to inboxes at remote locations and manage tasks in those inboxes. 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.

For more information about working with remote inboxes, see Enabling remote inboxes.

23. Prepopulate a target FSC for global data cache

Teamcenter provides a translator that you can run as an immediate, scheduled, or recurring task. You access the translator in the rich client. A structured context object (SCO) provides the information for prepopulating the FMS server cache (FSC) at Teamcenter sites that are not geographically near the database and therefore can use this feature to improve performance.

1. In My Teamcenter, select the SCO item you want to use to prepopulate your FSC.

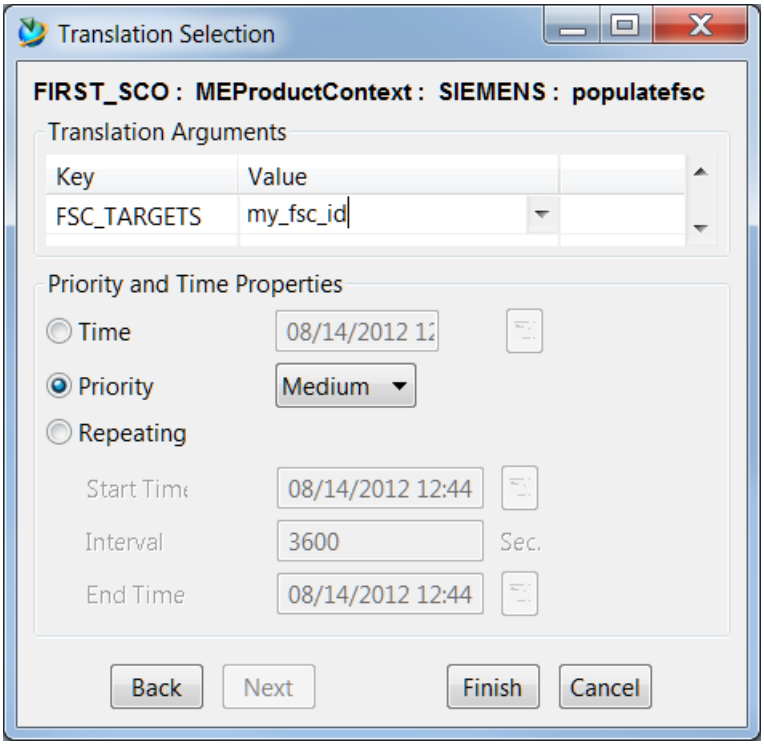
2. Choose **Translation→Translate**.

   Teamcenter displays the **Translation Selection** dialog box.

   The object you selected appears in the **Translation Services** list with **populatefsc** in the **Service** column.

3. Click **Finish** to initialize the service immediately for a one-time population. Otherwise, click **Next**.

   Teamcenter displays the **Translation Selection** dialog box.



   Set the desired properties for when and how often the services runs. The **FSC_TARGETS** identify the specific FSCs to populate. You can identify the FSCs using:

- A single **fscid** value or a list of comma-separated values.
- A single **populatetargets** value or a list of command separated values specified in the **exitfsc** elements in the FSC configuration file.
- A mix of comma-separated **fscid** and **populatetargets** values.
- A URL where the FSC is running in the form, **http://***host-name:port-number*, for example:

  ```
  http://mil4w001:4545
  ```

  mil4w001 is the name of the host where the FSC is running and **4545** is the port where the FSC is listening.

Caution:

The **fscid** and **populatetargets** values supplied for **FSC_TARGETS** must be predefined in the FMS configuration file in an **exitfsc** element, for example:

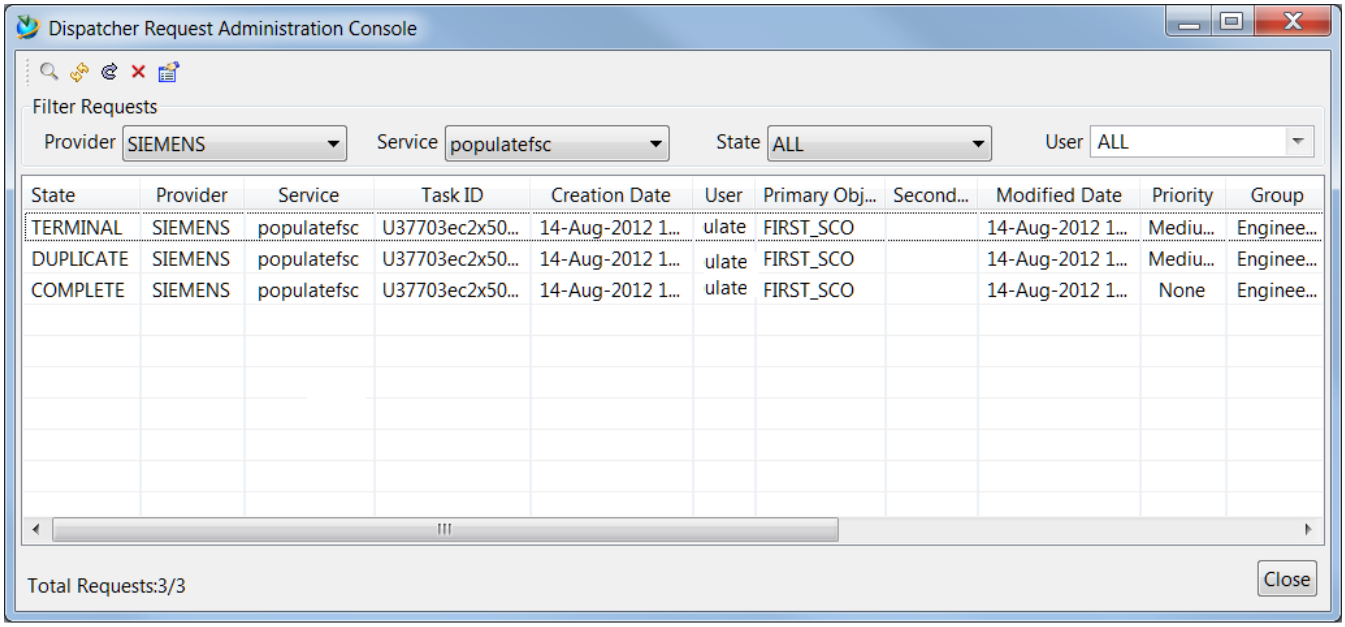
```
<exitfsc fscid="fsc1" populatetargetids="default,all"/>
```

A URL value is not required to be predefined in FMS.

The values in the field are not validated. If you type an incorrect value, the service action fails and writes an invalid target error in the FSC **syslog** file.

- To set the service to run at a future time, select the **Time** option and type the date and time in the adjacent box. You can also click the calendar button adjacent to the box and set the desired time a calendar dialog box.
- To set the service to run over a specific period and interval, select the **Repeating** option.
- The default option is **Priority**. Use this option to run the service immediately. Select the priority level from the list to set multiple services to run immediately with certain services run before others.

Use the **Dispatcher Request Administration Console** dialog box to view information about the **populatefsc** service. To access the dialog box, choose **Translation→Admin Console - All**.



A. Troubleshooting Multi-Site

Using Multi-Site troubleshooting information

This information is intended to help anyone supporting Multi-Site Collaboration. 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*.

If your organization is using Active Workspace, you can use Multi-Site Assistant to analyze and resolve item ownership, object ownership, and ID issues.

Although most of the examples presented in this reference are for Linux, 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** 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 *Fix an IDSM server connection 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. Also see *Setup verification* for checks you can use to diagnose a Multi-Site Collaboration problem when the system has been operational for some time or after a change (inadvertent or otherwise) in the configuration. For example, a site that has been successfully running suddenly develops a problem. This problem could be caused by changes in the **tc_profilevars** file, or the server node of an existing site may have been moved, but other sites were not notified.

Finding error codes and descriptions

All error codes are documented in the *Integration Toolkit Function Reference*. Error codes are grouped by module. For example, object import and export errors are listed within the OBJIO Errors module, data exchange errors are listed within the GMS Errors (Global Multi-Site) module, TC XML errors are listed in the TIE Errors (TCPLMXML Import/Export) module, and so forth.

To display a list of error messages:

1. Open the *Integration Toolkit Function Reference* (*ITK Function Reference*) in Support Center.

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

3. In the **Modules** page, scroll down to the appropriate module.

   For example, to see all object import and export (OBJIO) errors, which define many errors related to Multi-Site transfers, scroll to **OBJIO Errors** and click the link.

4. The error page displays all errors for that module. Error numbers are defined as *module-base-value* + *error-code*. The error base is defined in the EMH (Error Message Handler) Constants module.

   For example, the **OBJIO_unsupported_type** error has an error code of **EMH_OBJIO_error_base + 1**.

5. To determine the error base value for the selected module:

   a. Return to the **Modules** page.

   b. Scroll down to **EMH Constants** and click the link.

   c. The EMH Constants page displays the error base of each module.

      For example, the error base value of **EMH_OBJIO_error_base** is **41000**. Therefore, the error number for the **EPM_unsupported_type** error is the concatenation of the OBJIO modules error base (**41000**) and the error code **(1)**, creating an error code of **41001**.

6. To find descriptive text for an error, locate the error number in the module and click the error definition link.

   Many Multi-Site related errors are defined in the following modules.

Name	Error base	Value
SS Errors (System Services)	EMH_SS_error_base	1000
CXPOM Errors (CX Persistent Object Manager)	EMH_CXPOM_error_base	7000
WSO Errors (Workspace Object)	EMH_WSOPOM_error_base	27000
Reservation Errors	EMH_RESPOM_error_base	32000
OBJIO Errors (Object Import and Export)	EMH_OBJIOPOM_error_base	41000
Backup Import and Export Errors	EMH_BIERPOM_error_base	41100
ITEM Errors (Item)	EMH_ITEMPOM_error_base	48000
Publication Record Errors (Publication Record)	EMH_PUBRPOM_error_base	100000
ODS Errors (Object Directory Services)	EMH_ODSPOM_error_base	100100
IDSM Errors (Publication Record)	EMH_IDSMPOM_error_base	100200
IIR Errors (Item ID Registry)	EMH_IRRPOM_error_base	100300
POM Return Values and Tokens (Persistent Object Manager)	EMH_POMPOM_error_base	515000

Recovering from transfer errors

Replication errors

Specific **rules and restrictions** regarding object replication 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, these error recovery topics are intended to help you solve occasional problems. These techniques must not 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 Digital Industries Software recommends that you use checkpoints if your transaction has multiple target sites. The following are valid checkpoint arguments:

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.

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 primary object

When the primary 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 primary copies or overwriting the latest copy with an obsolete copy. This procedure must be performed by a user with Teamcenter administrator user privileges. 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 primary object

Under normal conditions, a primary 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 primary 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 primary object. To avoid synchronization errors, delete attached datasets first.

3. Manually delete the primary object.

Object ownership errors

Failures during 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.

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.

Determine objects requiring corrective action

Use either of the following methods to find objects flagged for corrective action:

- Search using the **__Objects_for_Site_Consistency** saved query in a Teamcenter rich client. You must have Teamcenter administrator privileges to use this query.
- Include the **report** function when running the **ensure_site_consistency** utility, for example:

```
ensure_site_consistency -f=report -search -report=recovery_candidates.txt
```

Run the **ensure_site_consistency** utility to correct the problem using the list of returned objects.

- To perform corrective action on a single object, enter:

```
ensure_site_consistency -f=recovery -item_id=Item123
```

- To perform corrective actions on all objects that require corrective actions and report the results, enter:

  ```
  ensure_site_consistency -f=recovery -search -report=recovered-objects-list.txt
  ```

  This approach is the recommended best practice, particularly in cases of multiple objects failing remote checkin.

- To perform corrective action on a list of items, enter:

  ```
  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 actions on all objects in a uniquely-named folder, enter:

  ```
  ensure_site_consistency -f=recovery -folder=unique-folder-name
  ```

  Using a folder is suitable for workspace objects that do not have unique IDs such as datasets and forms. Doing so is useful for failed remote check-ins of multiple objects when 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 failure or the unexpected termination of a process), Oracle will not immediately detect 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. **SQLNET.EXPIRE_TIME** is typically set to 10 to 15 minutes but twice the default value may pass 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 transfers of ownership 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 not transferred using the SST protocol that has ownership inconsistencies. 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 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.

Convert an item with mixed ownership to an 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*

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 an 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.

Teamcenter log files

Configuring the Multi-Site logging level

Teamcenter 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 of storage or performance problems.

Configure the logging level for all **TcServer** processes in your server pool using the **logger.properties** file. Use the **logging.logger.Teamcenter.Multisite** property to control the level of Multi-Site messages saved in the **syslog** file. Changes to the properties in the **logger.properties** file take effect after the Server Manager is restarted.

You can change the logging level at a remote site for the site's current session using the **dsa_util** utility. Also, you can determine the target IDSM host name using the correlation ID in the **syslog** file. Knowing the IDSM host name allows you to determine the appropriate **syslog** file for a remote site transaction. The correlation IDs are useful in troubleshooting transactions between specific sites. For publishing transactions, you can use the correlation ID to determine the ODS host name to locate the log files for this type of transaction. The following sample **syslog** file entry for a publish transaction shows the correlation IDs in bold:

```
INFO - 2011/7/13-13:54:28.611 UTC - pun6w388.04396.02.tcdba.00189 –
Remote API [ods11_1_publish_object] at Site [100002] Invoked -
Teamcenter.CMS.publication at
F:\workdir\t91_PL1\src\core\publication\ods_svc_proc.cxx (12866)

INFO - 2011/6/15-06:58:51.338 UTC - pun6w388.04396.02.tcdba.00189 –
Remote API [ods11_1_publish_object] at Site [100002] Returned ifail [0]
from server[puni6p163.00871.01.tcdba.0006]-
Teamcenter.CMS.publication at
F:\workdir\t91_PL1\src\core\publication\ods_svc_proc.cxx (12866)
```

The corresponding ODS **syslog** file entry shows the correlation IDs in bold:

```
INFO - 2011/6/15-06:58:51.291 UTC - pun6p163.00871.01.tcdba.0006 –
Client siteID [100001] [pun6w388.04396.02.tcdba.00189]
REQUEST [ods11_1_publish_object] BEGINS   - Teamcenter.CMS.publication
at
F:\workdir\t91_PL1\src\core\publication\ods_svc_proc.cxx(15520)

INFO - 2011/6/15-06:58:51.312 UTC - puni6p163.00871.01.tcdba.0006 –
Client site ID [100001] [pun6w388.04396.02.tcdba.00189]
REQUEST [ods11_1_publish_object] ENDS with ifail [0]   -
Teamcenter.CMS.publication at
F:\workdir\t91_PL1\src\core\publication\ods_svc_proc.cxx(15520)
```

You can configure the logging level in the four-tier architecture for the current Teamcenter session using a Java EE administrative interface. You can use the Teamcenter Management Console or a third-party JMX administration tool, such as JConsole. This type of change is in effect only for the current session and does not affect other **TcServer** processes in the server pool.

Configure the logging level using the Teamcenter Management Console

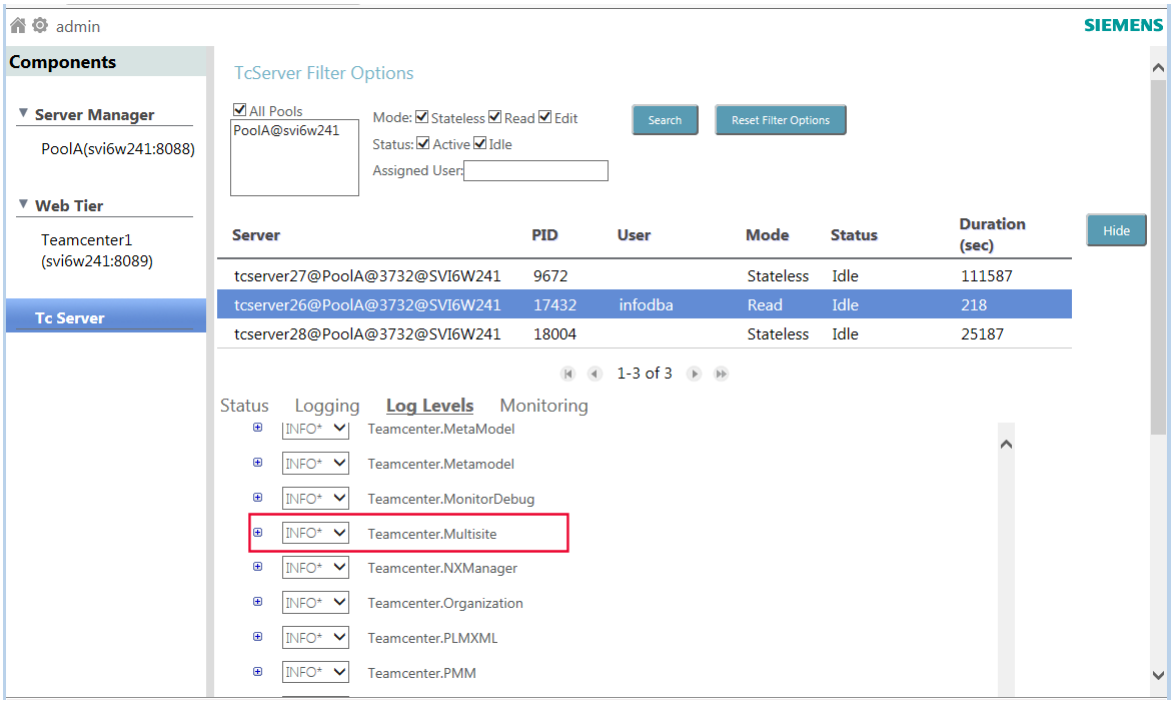
This procedures assumes you have the Teamcenter Management Console running.

1. Select **TcServer** and click the **Search** button to search for all active **TcServer** instances.

2. Select the server instance for which you want to change the logging level.

3. Click **Log Levels** and expand the list until you see **Teamcenter.Multisite**.

   This view shows the name of the logger and the current priority (level) set for the logger.



4. Select one of the following values in the priority box:

   - **TRACE**
   - **DEBUG**
   - **INFO**
   - **WARN**
   - **ERROR**
   - **FATAL**

   The logging level for the current Teamcenter session is set. No **TcServer** process restart is required.

Configuring remote site logging using the dsa_util utility

You can use the **dsa_util** utility with the **-f=set_logging_level** and **-level=***logging-value* arguments to temporarily set the logging level at a remote site for troubleshooting a Multi-Site transaction between two sites. You must have the **IDSM_dsa_sites_permitted_to_push_admin_data** preference set at the remote site with the site where you run the **dsa_util** utility included in the preference value. Because you can set the logging value temporarily (for the current session) of a remote site using the Java EE administrative interface, this procedure is intended for two-tier RPC environments.

Change a remote site's logging level

1. At the **Remote2** site, ensure that site is in the **IDSM_dsa_sites_permitted_to_push_admin_data** preference value.

   ```
   IDSM_dsa_sites_permitted_to_push_admin_data=Local1, ...
   ```

2. At the **Local1** site, open the **logger.properties** file and change the **Teamcenter.Multi-Site** preference value to **DEBUG**. This file is in the directory indicated by the *TC_LOGGER_CONFIGURATION* environment variable. By default, a copy is in the *TC_DATA* directory.

   ```
   Teamcenter.Multi-Site=DEBUG
   ```

3. Save the modified **logger.properties** file and restart the Server Manager.

4. In a Teamcenter environment command shell on the **Local1** site, type the following command.

   ```
   dsa_util u=tc-admin-name p=password -f=set_logging_level -level=DEBUG
   -site=Local1
   ```

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 than 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 consisting of a set of error codes and the corresponding meanings is displayed. 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, the remote site's IDSM process likely generated its own **syslog** file that may contain more helpful information.

For more basic debugging techniques, see *Debug remote import/export problems*.

Limit 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** preference value, and when the import state changes.

To limit the growth of the redo logs, set the **TC_RIMP_BG_prg_update_interval** 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.

Multi-Site Collaboration correlation ID

The Multi-Site correlation ID is categorized into the following types:

- **SOA generated correlation ID**

  The services oriented architecture (SOA) correlation ID is generated by the SOA layer when a client accesses the **TcServer** through SOA when the user logs on to Teamcenter through the rich client and when Multi-Site transactions are performed in a four-tier environment. The ID has the following format:

uscin6w034.	**42445.**	**01.**	**jdoe.**	**00247**
Client m/c of SOA connection	Thread ID	SOA connection index	User ID	Counter

- **Custom generated correlation ID**

  In the absence of an SOA layer, a custom correlation ID is generated internally by the Multi-Site code utilities (such as **data_share** and **data_sync**) and from the IDSM and ODS servers running in two-tier RPC architecture. The ID has the following format:

decol6s003.	**04980.**	**tcadm.**	**9**
Client m/c of SOA connection	Thread ID	User ID	Counter

When a proxy server is set up, the correlation ID points to a remote site server instead of the proxy server. In client and server syslog files, correlation IDs for the interacting client or server are present. In

the proxy environment, this behavior remains, and although an intermediate proxy exists, its correlation IDs are not present in the syslog file.

Correlation ID permutations

The following possible permutations illustrate the expected logging of the correlation IDs for a typical Multi-Site export transaction. Each correlation ID appears in bold to demonstrate how it can be used to trace a Multi-Site transaction end to end.

1. Four-tier HTTP environment

In a four-tier HTTP configuration, the **TcServer** processes act as IDSM/ODS servers, and an SOA connection is used to talk to these remote IDSM/ODS servers. An SOA connection generates its correlation ID by appending new information to the correlation ID that is passed from the local (source) site. If a transaction has a **uscin6w034.42445.01.jdoe.00247** correlation ID at the local site, the remote (target) site's SOA connection generates the correlation ID by appending **01.jdoe.00008** to the original correlation ID. Where **01** is the SOA connection index, **jdoe** is the user name used to create the SOA connection, and **00008** is a counter. The target site correlation ID becomes **uscin6w034.42445.01.jdoe.00247.01.jdoe.00008**.

The following example shows an export from a rich client. The source correlation ID follows the SOA generated correlation ID format because the transaction is run from the rich client.

Source site:

```
INFO  - 2011/11/15-17:54:20.717 UTC - uscin6w034.42445.01.jdoe.00247 -
Remote API[idsm11_distributed_app_1] at Site[300001019] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Invoked  -
Teamcenter.Multisite.publication at d:\tc91w1019_64\src\core\publication\
dist_itk.cxx(13131)

INFO  - 2011/11/15-17:54:20.744 UTC - uscin6w034.42445.01.jdoe.00247 -
Remote API[idsm11_distributed_app_1] at Site[300001019] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Returned ifail[0]
from server [uscin6w034.42445.01.jdoe.00247.jdoe.00008] -
Teamcenter.Multisite.publication at
d:\tc91w1019_64\src\core\publication\dist_itk.cxx(13131)
```

Target site:

```
INFO  - 2011/11/15-17:54:25.338 UTC -
uscin6w034.42445.01.jdoe.00247.jdoe.00008 -
Clientuscin6w034.42445.01.jdoe.00247] SiteID[-2090444906]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] BEGINS -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1019_sol/src/core/publication/idsm_svc_proc.cxx(16125)

INFO  - 2011/11/15-17:54:25.342 UTC -
uscin6w034.42445.01.jdoe.00247.jdoe.00008 -
Client[uscin6w034.42445.01.jdoe.00247] SiteID[-2090444906]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP] appcode[1
```

```
DIST_IDSM_exchange_supported_feature_set_op] ENDS with ifail [0] -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1019_sol/src/core/publication/idsm_svc_proc.cxx(16300)
```

The following example shows an export using the Multi-Site **data_share** utility. The source correlation ID follows the custom generated correlation ID format.

Source site:

```
INFO  - 2011/11/15-16:00:52.842 UTC - decol6s003.plm.eds.com.952592960.tcadm.1 -
Remote API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Invoked  -
Teamcenter.Multisite.publication at /plm/cynasew/tce_iproot/units/
tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)

INFO  - 2011/11/15-16:00:52.843 UTC - decol6s003.plm.eds.com.952592960.tcadm.1 -
Remote API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Returned ifail[0]
from server [decol6s003.plm.eds.com.952592960.tcadm.1.01.tcadm.00008] -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units
/tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)
```

Target site:

```
INFO  - 2011/11/15-17:54:25.338 UTC -
decol6s003.plm.eds.com.952592960.tcadm.1.01.tcadm.00008 -
Client[decol6s003.plm.eds.com.952592960.tcadm.1] SiteID[-2090444906]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op]
BEGINS - Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units
/tc91_week_1019_sol/src/core/publication/idsm_svc_proc.cxx(16125)

INFO  - 2011/11/15-17:54:25.342 UTC -
decol6s003.plm.eds.com.952592960.tcadm.1.01.tcadm.00008-
Client[decol6s003.plm.eds.com.952592960.tcadm.1]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP]
appcode[1 DIST_IDSM_exchange_supported_feature_set_op] ENDS with ifail [0] -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1019_sol/src/core/publication/idsm_svc_proc.cxx(16300)
```

2. Two-tier RPC environment

In a two-tier RPC environment, an SOA connection is not used for communication between the source and target site, therefore, both the sites generate independent correlation IDs.

The following example shows an export from a rich client. The source correlation ID follows the SOA generated correlation ID format because the transaction is run from the rich client.

Source site:

```
INFO  - 2011/11/15-15:25:49.085 UTC -
chm6w409.53827.01.jdoe.00150 - Remote
API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Invoked  -
```

```
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)

INFO  - 2011/11/15-15:25:49.086 UTC - chm6w409.53827.01.jdoe.00150 - Remote
API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Returned ifail[0]
from server [day6s003.04980.zztcadm.3] -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)
```

Target site:

```
INFO  - 2011/11/15-15:25:44.608 UTC - dayt6s003.04980.zztcadm.3 -
Client[chm6w409.53827.01.jdoe.00150] SiteID[-2091386568]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op]
BEGINS - Teamcenter.Multisite.publication at
D:\workdir\tc911005win64\src\core\publication\idsm_svc_proc.cxx(16119)

INFO  - 2011/11/15-15:25:44.608 UTC - day6s003.04980.zztcadm.3 -
Client[cmh6w409.53827.01.jdoe.00150] SiteID[-2091386568]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP]
appcode[1 DIST_IDSM_exchange_supported_feature_set_op]
ENDS with ifail [0] - Teamcenter.Multisite.publication at
D:\workdir\tc911005win64\src\core\publication\idsm_svc_proc.cxx(16294)
```

The following example shows an export using the Multi-Site **data_share** utility. The source correlation ID follows the custom generated correlation ID format.

Source site:

```
INFO  - 2011/11/15-16:00:52.842 UTC -
decol6s003.net.plm.eds.com.952592960.tcadm.1 - Remote
API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Invoked  -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units/
tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)

INFO  - 2011/11/15-16:00:52.843 UTC -
decol6s003.net.plm.eds.com.952592960.tcadm.1 - Remote
API[idsm11_distributed_app_1] at Site[-2091674577] application[IDSM_APP]
app_op_code[1 DIST_IDSM_exchange_supported_feature_set_op] Returned ifail[0]
from server [day6s003.04820.zztcadm.4] -
Teamcenter.Multisite.publication at /plm/cynas/tce_iproot/units
/tc91_week_1005_lnx64/src/core/publication/dist_itk.cxx(13131)
```

Target site:

```
INFO  - 2011/11/15-16:00:48.318 UTC - day6s003.04820.zztcadm.4 -
Client[decol6s003.net.plm.eds.com.952592960.tcadm.1] SiteID[-2091386568]
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP] app_op_code
[1 DIST_IDSM_exchange_supported_feature_set_op] BEGINS - Teamcenter.Multisite.publication
at D:\workdir\tc911005win64\src\core\publication\idsm_svc_proc.cxx(16119)

INFO  - 2011/11/15-16:00:48.319 UTC - day6s003.04820.yytcadm.4 -
Client[decol6s003.net.plm.eds.com.952592960.tcadm.1] SiteID[-2091386568]
```

```
REQUEST[idsm11_distributed_app_1_svc] app[IDSM_APP] appcode
[1 DIST_IDSM_exchange_supported_feature_set_op] ENDS with ifail [0] -
Teamcenter.Multisite.publication at D:\workdir\tc911005win64\src\core
\publication\idsm_svc_proc.cxx(16294)
```

3. RPC proxy environment

In a proxy server [1] environment, an intermediate server acts as a proxy. The intermediate proxy server generates its own correlation ID. However, this correlation ID does not play any role in the end-to-end Multi-Site transaction. A proxy server simply transfers the request from the source site to the target site and vice versa.

Source site:

```
INFO  - 2011/12/15-07:16:37.860 UTC - ind6w1232.04112.tcadmin.1 -
Remote API[idsm11_export_status_1] at Site[100001] Invoked  -
Teamcenter.Multisite.publication at D:\tc91w1212\src\core
\publication\idsm_itk.cxx(10071)

INFO  - 2011/12/15-07:16:39.607 UTC - ind6w1232.04112.tcadmin.11 -
Remote API[idsm11_export_status_1] at Site[100001] Returned ifail[0] from server
[ind6w1232.03956.tcadmin.8] - Teamcenter.Multisite.publication at
D:\tc91w1212\src\core\publication\idsm_itk.cxx(10071
```

To find the server proxy logs on the client side for a transaction:

1. Get the server correlation ID from the **syslog** file.

2. Go to the proxy logs location. The proxy host name is specified as the remote site's node name in the organization panel site definition at the client site.

3. In the proxy logs, look for the same server correlation ID that is needed on the client side.

   The log containing the server-side log is the corresponding client-side log.

Use the same process in the reverse direction for identifying client proxy logs.

Proxy site:

```
INFO  - 2011/12/15-07:16:37.860 UTC - cvg6s282.05832.UnknownClient.8 -
Client[ind6w1232.04112.tcadmin.1] SiteID[100002] REQUEST[idsm11_export_status_1_proxy]
to PROXY BEGINS - Teamcenter.Multisite.publication at
D:\tc91w1207_64\src\core\publication\idsm_svc_proc.cxx(13605)

INFO  - 2011/12/15-07:16:37.860 UTC - cvg6s282.05832.UnknownClient.8 -
Remote API[idsm11_export_status_1] via PROXY at Site[100001] Invoked  -
Teamcenter.Multisite.publication at D:\tc91w1207_64\src\core\publication\
idsm_svc_proc.cxx(13608)
```

1 The server and client receive correlation IDs for each other, and not for the proxy server. Providing proxy server correlation IDs may be implemented in a future release.

```
INFO  - 2011/12/15-07:16:39.595 UTC - cvg6s282.05832.UnknownClient.8 -
Remote API[idsm11_export_status_1] via PROXY at Site[100001] Returned ifail[0]
from server [ind6w1232.03956.tcadmin.8] -
Teamcenter.Multisite.publication at D:\tc91w1207_64\src\core\publication\
idsm_svc_proc.cxx(13608)

INFO  - 2011/12/15-07:16:39.595 UTC - cvg6s282.05832.UnknownClient.8 -
Client[ind6w1232.04112.tcadmin.1] SiteID[100002]
REQUEST[idsm11_export_status_1_proxy] to PROXY ENDS with ifail[0] -
Teamcenter.Multisite.publication at D:\tc91w1207_64\src\core\publication\
idsm_svc_proc.cxx(13610)
```

Target site:

```
INFO  - 2011/12/15-07:16:39.607 UTC - ind6w1232.03956.tcadmin.8 -
Client[ind6w1232.04112.tcadmin.1] SiteID[100002] REQUEST[idsm11_export_status_1_svc]
BEGINS - Teamcenter.Multisite.publication at
D:\tc91w1212\src\core\publication\idsm_svc_proc.cxx(17752)

INFO  - 2011/12/15-07:16:39.607 UTC - ind6w1232.03956.tcadmin.8 -
Client[ind6w1232.04112.tcadmin.1] SiteID[100002]
REQUEST[idsm11_export_status_1_svc] ENDS with ifail [0] -
Teamcenter.Multisite.publication at
D:\tc91w1212\src\core\publication\idsm_svc_proc.cxx(18032)
```

Common installation-related problems

Fix an IDSM server connection 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. On Linux, check the RPC connection to the remote sites IDSM server node. (The ability to check if the listener is ready is not available on Windows.)

   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
   ```

   In this example, 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

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.

  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 the IDSM **syslog** file.

   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.

Fix an ODS server connection 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 (Linux).

   (The ability to check if the listener is ready is not available on Windows.)

   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
```

In this example, 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**.

4. Check that the ODS server process is running.

   You must log on 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 **$TC_ROOT/bin/run_tc_ods** and the other with the path name of the ODS executable, the expansion of **$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.

Fix an ODS returning 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:

1. Check the ODS syslog to debug the **syslog**.

   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.

2. Generate ODS log files to generate complete log files.

   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.

Fix a server 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.

  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**. Send the log files in with your problem report.

Common import/export problems

Debug remote import/export problems

In most cases, when a remote import operation fails, error codes and messages are sufficient to help resolve the problem. For example, an error message may indicate the lack of privilege to import a given remote object; or when reimporting an object, the error may 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 troubleshooting topic. Rather, this topic discusses errors where the resolution is not straightforward, even for a system administrator who is trying to help a user resolve an import problem.

> Note:
>
> This debugging 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.

The debugging 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, use the following actions to resolve the problem:

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 Digital Industries 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.

Fix an 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 time stamp is the cryptic string that is part of the transmit file name. To ensure you have the 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 Tc-admin-user password group
   ```

   Where *Tc-admin-user*, *password*, and *group* are the credentials for a user with Teamcenter administrative privileges.

   Distribute the new transmit file to the other sites, particularly 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 *Debug remote import/export 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 Digital Industries 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 primary 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 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.

After running one or both of these utilities, the site ownership of all objects in the item should be consistent. If it is a primary copy, try exporting it without site ownership transfer. If the export succeeds, this confirms the success of the recovery.

Data synchronization does not change ownership at replica site

When you change user/group ownership of a primary object and subsequently run the **data_sync** utility to apply the change on the replica objects, object ownership changes may not occur. There are several possible causes:

- The primary object owner does not exist at the replica site. To change the ownership of the replica, the new owner must exist as the same user in the same group as at the owning site.

- The **TC_preserve_original_owner_on_sync** preference value is set to **FALSE** or is not defined at the replica site. This preference controls whether or not an object's owning user is updated during synchronization.

If a change in the primary object owner's group does not cause a corresponding change in the group of the replica after synchronization, ensure the **TC_retain_group_on_import** preference is set to **TRUE**. This preference overrides the **TC_preserve_original_owner_on_sync** setting.

If you set the **TC_retain_group_on_import** preference to **TRUE**, the owing user must be a the member under that group to allow the change during synchronization.

Configure a Teamcenter UTF-8 execution environment on Linux

If you are running the IDSM server process on Linux, 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 Linux systems when the IDSM daemon is not started with identical Teamcenter execution environment settings. By default, the IDSM server process is started by the Linux **idsminetd** daemon in the C locale.

To establish a Linux 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
```

Special characters not displayed properly on Windows client

There is a limitation within the Teamcenter localization architecture with respect to the import/export report. When a report is viewed from a Windows client and the corresponding server is running on Linux, the client may not display the special characters of foreign locale correctly. To avoid this issue, whenever the Teamcenter server is running on a Linux machine, the client must also be launched from the Linux environment using a tool such as a common desktop environment.

Resolving and preventing duplicate 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.

> Note:
>
> Cases of duplicate item IDs result from one of six possible causes. Therefore, it is important that you use the **data_share** utility to find cases of duplicate IDs and help you determine the cause.

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 enabling the central item registry at Multi-Site Collaboration sites. 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.

Enabling the central item registry

You can 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 within the Multi-Site Collaboration federation. The registry is the **ItemIdRegistry** table containing the set of all item IDs created within the Multi-Site Collaboration federation.

Enable the central item registry functionality by setting the following preferences:

- Set the **ITEM_id_registry** 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** preference to define the site of the registry.

- Set the **ITEM_id_always_register_on_creation** 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** 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** preference to determine if item IDs are automatically unregistered when items are deleted or the item ID is changed.

If you have set the preferences at your site to use the central item registry, Teamcenter checks the central item registry for a duplicate ID when you click **Finish** in the **New Item** wizard. If the ID already exists, the new item is not created. To avoid this, configure different naming rules for item IDs at each site participating in your Multi-Site federation and use the **Assign** button to set the ID.

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** preference.

Resolving identical items with different unique IDs

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 primary copy. Import the other item with transfer ownership for the forms and datasets that do not exist on the primary item. Create new revisions on the primary item for the objects imported.

Resolving entirely different items with the same ID

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.

Teamcenter services on Window systems

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 Linux and are normally started at boot up time.

The IDSM on Linux 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 IDSM server on demand.

Determine if Windows 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 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 *Fix an ODS server connection error*, to perform the RPC connection test for an ODS server.

- Follow the steps in *Fix an IDSM server connection error*, to perform the RPC connections test for an IDSM server. (On Windows, Teamcenter is shipped with a Windows version of the **rpcinfo** program located in the **%TC_BIN%** directory.)

Multi-Site and Security Services compatibility

Multi-Site remote procedure call mode does not support Security Services

When Multi-Site is using remote procedure call (RPC) mode and Teamcenter is installed in the Security Services enabled mode, the Integrated Distributed Services Manager (IDSM) and Object Directory Services (ODS) processes fail to respond because access is denied.

To allow Multi-Site to work, you must disable Security Services for the IDSM and ODS processes in your Teamcenter environment. You can enable Security Services for all other components.

If you want all components on the same machine, associate a new *TC_DATA* directory with the IDSM and ODS processes that is not Security Services enabled.

Disable Security Services for the IDSM and ODS processes

1. Duplicate the *TC_DATA* directory by creating the **${TC_DATA}_nonsso/** or **%TC_DATA%_nonsso** directory (depending on the platform).

2. Unset all Security Services related variables defined in the **${TC_DATA}_nonsso/tc_profilevars** or **%TC_DATA%_nonsso\tc_profilevars.bat** script. The three variables that must be removed or reset are **TC_SSO_APP_ID**, **TC_SSO_SERVICE**, and **TC_SSO_LOGIN_URL**.

   - Windows example:

     ```
     rem set TC_SSO_APP_ID=Tc8S17
     rem set TC_SSO_SERVICE=http://
     svli6011v03.net.plm.eds.com:7105/ssoService8
     rem set TC_SSO_LOGIN_URL=http://
     svli6011v03.net.plm.eds.com:7105/ssoLogin8
     ```

   - Linux example:

     ```
     #TC_SSO_APP_ID=Tc8S19; export TC_SSO_APP_ID
     #TC_SSO_SERVICE=http://
     svli6011v03.net.plm.eds.com:7105/ssoService8;
     export TC_SSO_SERVICE
     #TC_SSO_LOGIN_URL=http://
     svli6011v03.net.plm.eds.com:7105/ssoLogin8;
     #export TC_SSO_LOGIN_URL
     ```

3. Set the *TC_DATA* variable to point to the newly created **${TC_DATA}_nonsso** (Linux) or **%TC_DATA%_nonsso** (Windows) directory in the following files.

- Windows:

  *TC_ROOT***bin\run_tc_ods.bat**
  *TC_ROOT***bin\run_tc_idsm.bat**

- Linux:

  TC_ROOT/**bin/run_tc_ods.sh**
  TC_ROOT/**bin/run_tc_idsm.sh**

Save the file and restart the IDSM and ODS processes.

Error recovery procedures

For various reasons, it is possible that the primary object 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 primary 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 with the **-f=cancel_remote_co** option to cancel the remote checkout. Then, log on to the replica site and perform a remote checkout on the replica containing the changes that were made to it and check the object back in.

If the replica is checked out but the primary copy is not, this condition is most likely caused by someone at the owning site forcibly canceling the remote checkout on the primary copy. Use the **data_share** utility at the replica side with the **-f=cancel_replica_co** option to cancel the replica check out. Then, 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.