



TEAMCENTER

Teamcenter Site Consolidation

Teamcenter 2412

Unpublished work. © 2025 Siemens

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

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

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

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

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

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

About Siemens Digital Industries Software

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

Support Center: support.sw.siemens.com

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

Contents

Getting started with site consolidation

Site consolidation	1-1
Understanding the site consolidation process	1-1
Team, business, and technical drivers	1-3
Setting goals	1-3
Developing a high-level plan	1-4
Reviewing your goals and plan	1-4
Source and target site requirements	1-5
High-level technical considerations for site consolidation	1-6
Technical and compatibility requirements	1-6
Physical infrastructure issues	1-7
Performance monitoring	1-9
Client performance considerations	1-9
Enterprise server tier issues	1-10
Database server considerations	1-10
Sites	1-11

Basic concepts for using site consolidation

Process definition assumptions	2-1
Site consolidation phases	2-2
Preparation phase	2-2
Execution phase	2-3
Cleanup phase	2-4
Planning cleanup and volume relocation activities	2-5
Teamcenter Integration Framework orchestration	2-6
Suppress the Teamcenter Integration Framework export command	2-9
Site consolidation tools and utilities	2-9

Preparing for site consolidation

Deploying Teamcenter components	3-1
Upgrade the ACCT_TABLE database table	3-2
Maintain data synchronization through site consolidation	3-3
Making site definitions compatible	3-4
Rationalizing site definitions	3-4
Consolidating administration data	3-4
Process for consolidating administrative data	3-5
Rule trees	3-6
Detecting and correcting data issues at the source and target sites	3-17
Schema compatibility	3-19
Planning increments	3-25
Data partitions	3-25
Creating partitions	3-25

Developing a detailed transfer plan	3-26
Improving plm_report_extract utility performance	3-28
Run the plm_report_extract utility with lightweight closure rules	3-29
Create a report with units of measure	3-30
Detecting and repairing data anomalies	3-30
Detection and repair tools	3-30
Replace invalid characters	3-31
Detect and correct any source site references to CFM configuration object data	3-32
Missing file data at the source site	3-32
Managing intersite conditions	3-33

Execution phase activities

Execution phase increments	4-1
Replicating data and changing ownership in a typical execution phase increment	4-1
Consolidating using multiple execution phase increments	4-2
Replication phase	4-2
Replication phase activities	4-2
Replication phase orchestration process	4-5
Time critical period	4-6
Time critical period activities	4-6
Performance considerations during the time critical period	4-8
Disabling user access and backing up data	4-9
Performing final replication and synchronization	4-9
Performing validation	4-9
Change object ownership	4-9
Sequencing change ownership tasks automatically	4-10

Cleanup phase activities

Determining when to perform cleanup activities	5-1
Resolving issues after moving consolidation information	5-1
Cleaning up the target site	5-2
Cleaning up the ODS	5-3
Cleaning up third sites	5-3
Cleaning up the source site	5-5
Determining final cleanup steps	5-5

Site consolidation utilities

Consolidation from Oracle to Microsoft SQL Server

Environment and general information	B-1
Preparation phase process example	B-2
Configure the source site	B-2
Configure the target site	B-3
Prepare a transfer plan	B-5

Shared FMS network

Prerequisites for a shared FMS network	C-1
Key FMS configuration values before sharing	C-1
FMS configuration files before sharing	C-2
Key FMS configuration values after sharing	C-4
Configure shared FMS	C-5
Install the shared network configuration files	C-7
Create shared volumes	C-10

Auditing affected sites

Using audit components	D-1
Considering infrastructure issues	D-2
Recovering from failures during consolidation activities	D-3
Planning for recovery	D-3
Recovery activities	D-3
Preparation phase activities	D-4
Execution phase activities	D-4
Replication, fast synchronization, and final preparation for the time critical period	D-5
Time critical period	D-5
Cleanup and retirement	D-5

Performance tuning

Using database statistics	E-1
Index performance examples	E-1
Run 1: wildcard (*) in command argument	E-1
Run 2: no wildcard (*) in command argument	E-1
Run 3: no wildcard (*) in command argument, added db index for export records	E-2
Run 4: no wildcard (*) in command argument; db index for export records; added db index for variant expressions	E-2
Setting Oracle SDU/TDU values	E-3
Setting the Oracle tcp.nodelay parameter	E-3
Oracle System Global Area	E-3
Oracle Program Global Area	E-4
Additional Oracle parameters that impact database performance	E-4
Automatic Database Diagnostic Monitor	E-5
Networking	E-6
File system	E-6

Using log files

Log file names and locations	F-1
Utility debug options and default file locations	F-1
TIE_DEBUG environment variable	F-2
Sample run of tcxml_import low-level	F-4
Orchestration data transfer log files	F-7

Orchestration change ownership log files	F-8
Teamcenter Integration Framework tables cleanup	F-9

Frequently asked questions

Can site consolidation utilities be used with 4th Generation Design data?	G-1
How are unmodifiable 4th Generation Design properties handled during import?	G-1
What can be done if the replication process aborts because special characters are encountered during mapping?	G-3
What does the term site definition data include, which must be addressed during the preparation phase?	G-3
How are scalability and performance impacted by island size?	G-4
Why do we need the <code>fix_release_status</code> , <code>clean_up_shared_objects</code> , and <code>sitcons_fix_ixr</code> utilities?	G-4
What are closure rules?	G-5
What happens when a network loss or outage happens during site consolidation?	G-5
What can be done if replication aborts because <code>sitcons_replicate_mgr</code> mapping fails when special characters, such as illegal XML 1.0 characters, are encountered?	G-5
How can I override the default delimiter in TC XML for string variable length arrays (VLAs)?	G-6
When you transfer ownership of data having subscription, does the subscription get transferred along with it?	G-6
How do you determine whether unreferenced objects are orphaned?	G-6
How do you transfer objects not referenced by an item but by a folder?	G-6
How do I configure the Teamcenter Integration Framework URL for site consolidation	G-6
Can the file client cache (FCC) recognize cached data if the FMS ticket is from a different server?	G-7
What happens to replicated files when a site is consolidated after a period of data exchange?	G-7
Can we use volumes located near remote (former source) sites to increase operational speed?	G-7
When does the database fail?	G-7
What is the process of deleting files from the source site if files on the target site are up to date?	G-7
What impact do the database tuning guidelines have on the production database?	G-8
What can be done when using the <code>sitcons_replicate_mgr</code> and there is failure to connect?	G-8
What happens when an object's User/Group profile is not found in the target database during import?	G-9
How do I identify the failure point when the accountability table trigger creation fails without error information?	G-9

Why do I get an FSC proxy error when I access a volume in a shared FMS environment? _____	G-9
How do site consolidation closure rules handle incremental change elements? _____	G-10
How do I resolve an import failed error that indicates the last_mod_user was not found in the database for a workspace object? _____	G-10
Why is my custom form object exported in a different island by itself? _____	G-10
Why does Teamcenter Integration Framework fail on import and display an XML/XSL file for import/export is invalid error? _____	G-11
Why does the sitcons_replicate_mgr utility not submit some requests? —	G-12
How do I transfer Audit Manager logs to the target site? _____	G-12

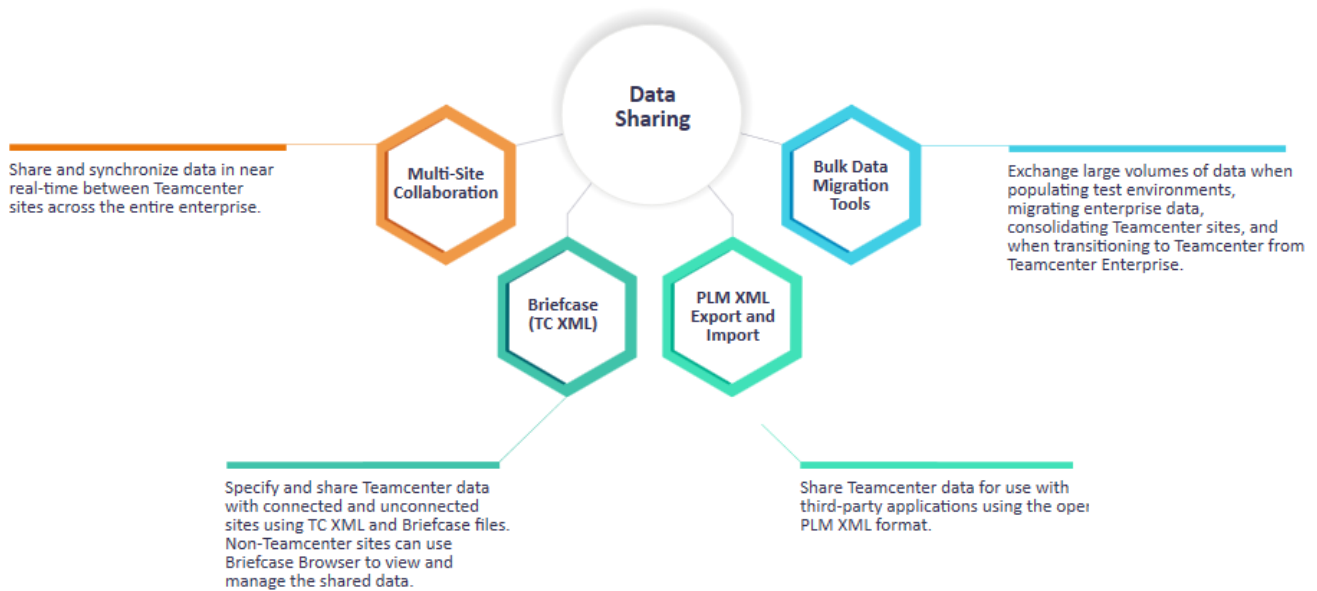


1. Getting started with site consolidation


Site consolidation

As the requirements of your organization change, you may find the need to reduce the number of Teamcenter sites in operation. Site consolidation is a collection of tools and processes that enable you to relocate data from sites before retiring those sites.

Site consolidation is one of the bulk data migration tools Teamcenter offers to meet different data sharing needs.



Where do I go from here?

 Administrator	
Understanding site consolidation.	See Overview of the site consolidation process .
Preparing for site consolidation.	See Team, business, and technical drivers .
Reviewing the tools employed during site consolidation.	See Site consolidation tools and utilities .

Understanding the site consolidation process

The site consolidation process, which reduces the number of Teamcenter sites in operation in an enterprise multiple-site environment, requires extensive planning and uses a variety of Siemens Digital Industries Software tools.

Warning:

Some of the tools and processes used in this process can cause data corruption or data loss if they are not used or performed properly.

Because of the potential for damage to your enterprise data, you must open a support case on Support Center to obtain a key for the use of site consolidation tools and to 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 consolidate sites.

A typical large enterprise may reduce a dozen sites to one or two sites.

The site consolidation process is organized in three distinctive phases:

- **Preparation**

During the *preparation* phase, you collect data, do in-depth analysis, and perform specific readiness steps to ensure site definition, product, and volume data is ready for consolidation.

- **Execution**

During the *execution* phase, you move product data from the source to the target site. It consists of multiple increments based on the source site size and the business drivers for the consolidation activity.

- **Cleanup**

During the *cleanup* phase, you do final cleanup, retirement, and, optionally, decommissioning of the source site, including activities such as volume data relocation, ownership conflict resolution, and Object Directory Service (ODS) configuration/cleanup.

Caution:

Site consolidation procedures are complex and require precision planning and execution.

- This document, the site consolidation tools, and all relevant supporting material do not guarantee success prior to, during, or post consolidation.
- A well-planned methodology developed by an experienced system integrator is essential.

The site consolidation tools and detailed documentation are protected by license to ensure that a proper qualification is done through the Siemens Digital Industries Software Global Sales and Services organization in cooperation with the Siemens Digital Industries Software Research and Development organization.

Team, business, and technical drivers

Site consolidation requires a multidisciplinary team versed in the business, legal, infrastructure, data management, and end-user issues impacted by the effort.

- Many business drivers can create a need for consolidation activity, such as infrastructure cost reduction, Information Technology (IT) centralization, and business acquisition.
- End-user performance considerations, such as network latency, proximity to volume data, and global data sharing patterns, can create the need for consolidation.
- OEM, supplier, joint-venture situations require careful planning and analysis from business and legal perspectives.
- Business initiatives that drive global data replication scenarios must be considered for successful timing and coordinated data movement.
- Active production data growth during the preparation phase, which could take several weeks or months, must be anticipated by strategic planning.

The active growth for source sites and target sites, which impacts server and storage capacity planning, must be determined far enough in advance of consolidation to account for procurement lead time.

- All involved sites (source, target, and third sites) must be operating at the same appropriately licensed Teamcenter version level, which includes the site consolidation tools. The involved third sites are those that have shared Teamcenter data with the source site. The site consolidation tools must be run at these sites during the cleanup phase.

To identify the business and technical drivers, and to determine the overall feasibility of site consolidation, you can use the site consolidation tools to help collect the data from identified source and target locations.

Setting goals

Prior to starting site consolidation work, conduct a planning process to set goals.

- Identify sites that align to business objectives (legal, financial, performance).
- Determine business process improvement goals, such as global data collaboration patterns.
- Specify cost reduction targets.
- Determine performance benchmark targets for user acceptance.
- Specify execution milestones.

Developing a high-level plan

After setting goals for site consolidation work, develop a high-level plan.

- Create a time line to align to the executive milestones.
- Establish a budget.
- List required hardware resources.
- List required software resources.
- List required staffing resources.
- Determine business and legal compliance (ITAR/EC/IP) requirements.
- Specify technical audit information required for inputs into the high-level plan development.

You can leverage site consolidation tools to help collect this input.

Reviewing your goals and plan

After setting goals and developing a high-level plan, review the goals and plan in light of the following considerations:

- Multiple source sites may need to be consolidated to one target site, but execution takes place one source site at a time.
- Sites being consolidated must all be specified, such as the source, target, and involved other sites.
- The reasons for the original separation of these sites, which must be consolidated and addressed, can include:
 - Performance
 - Security
 - Business or legal
 - Infrastructure
 - Software
 - Support

- After candidate sites are selected, analyze impacts in the following areas:

- User loading
- Data sharing implications

For example, you may need to prepare for new user loading or changed site sharing patterns.

- Functionality and software

For example, you may need Teamcenter upgrades and additional integrated or interfaced applications such as enterprise resource planning (ERP), supplier relationship management (SRM), source code management (SCM), and so on.

- Performance commitments
- Infrastructure

For example, you may need to upgrade servers, network equipment, and storage.

- Tailoring

For example, you may need configuration and software changes at the target site based on source site considerations.

- Support requirements

For example, you may need to inform help desk and system administration personnel.

- IT infrastructure

Source and target site requirements

Ensure the planning process addresses the following requirements:

- The data, users, and processes of the source site can be moved from the source to the target site in partitions.
- The source and target sites are in production operation on the same release of Teamcenter.
- The source and target sites can be made to have consistent site definition data through manual changes or metadata mapping.
- The target site contains the user that owns the object at the source site. Otherwise, the object is owned at the target site by the user that performs the import.

- The source and target sites have compatible character representations at the database level.
- The target site supports all locales for the data transferred from the source site. Otherwise, data for the unsupported locales is lost with no error reported.
- No metadata language translation (such as LOVs) of volume data is required.

High-level technical considerations for site consolidation

Technical and compatibility requirements

The technical considerations for site consolidation include deployment of new Teamcenter components, the data model, and, potentially, incremental infrastructure.

Consolidation using the Siemens Digital Industries Software provided site consolidation tools requires a shared File Management System (FMS) network. The source and target site definition data (administrative and schema) also must be compatible. You must also consider if additional infrastructure is required beyond the infrastructure in place to support normal production usage for the concurrent Teamcenter deployment.

Additional compatibility requirements include:

- Siemens Digital Industries Software recommends that you deploy the item registry capability.

The item registry capability can prevent an import failure caused by items with the same item ID and different unique identifiers (UIDs) in the source and target systems. During the preparation phase, the presence of items with the same item ID but different UIDs is checked by the **plm_report_constraint** utility analysis. However, because the replication period can occur over time, users may re-create the failure condition after the preparation phase cleanup, unless item registry is deployed.

Do one of the following:

- If you deploy the item registry functionality capability, set up a separate node for the ODS. That node can either be connected using RPC or HTTP, but for HTTP, some additional settings are required.
- If you do not deploy the item registry functionality capability, set the **ITEM_id_registry** preference value to **FALSE** so item creation can coexist with site consolidation orchestration.
- The Teamcenter Integration Framework orchestration component provided by groovy scripts allows you to automate the multistep processes required for replication, synchronization, and ownership change of product data.

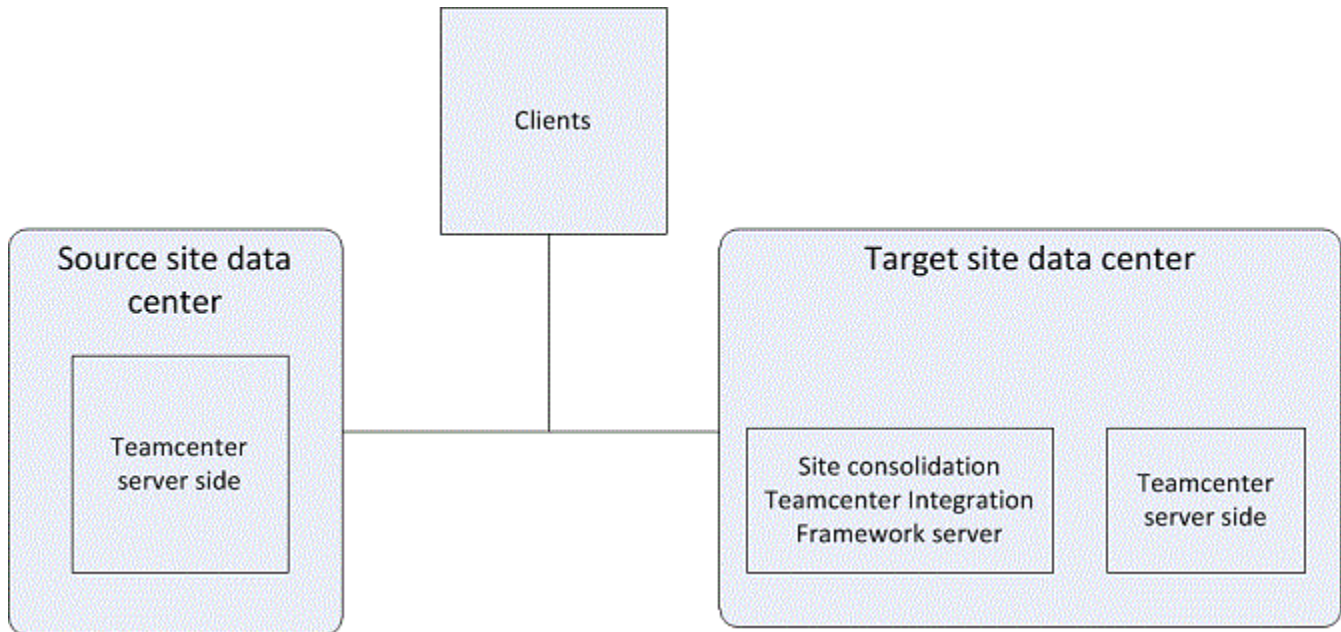
Before using the Teamcenter Integration Framework, consider the following points:

- While automatically performing a complex sequence of steps, Teamcenter Integration Framework determines log file names and locations, and limits your ability to easily examine intermediate outputs.
- The availability of a monitoring user interface and the automatic notification of step status in orchestration does not relieve you of the responsibility to verify success separately.
- If a task reports failure, orchestration automatically rolls back the export to a prior step.
- Teamcenter Integration Framework employs a Java Virtual Machine (JVM) mapping step. The JVM has a relatively low memory size, so orchestration may require you to create more partitions than you would with manual processing.
- Infrastructure demands may be higher for Teamcenter Integration Framework because **orchestration** uses **TcServers** at both the source and target sites.
- The **SITCONS_AUTH_KEY** environment variable must be set to a valid key value to run the following site consolidation utilities. Open a support case on Support Center to get this license key.
 - **sitcons_accountability_chk**
 - **sitcons_extract_shared_vols**
 - **sitcons_fix_ixr**
 - **sitcons_gen_shared_vols**
 - **sitcons_replicate_mgr**
 - **sitcons_user_folders**
 - **sitcons_xfer_owner_mgr**

Physical infrastructure issues

Physical infrastructure, such as hardware platforms, network connecting the sites, and storage, must be appropriately sized, correctly functioning, and available to the site consolidation tools. Consolidation infrastructure sizing depends on performance and scalability requirements specified for the consolidation effort.

The following diagram shows a high-level overview of typical deployment architecture for site consolidation using the Teamcenter Integration Framework orchestration component.

**Note:**

Third sites are involved in site consolidation only during the cleanup phase. During cleanup, objects that the source site previously owned and that the third site is aware of must be updated to reflect the correct site ownership.

While consolidation activities can be performed manually, the orchestration component ensures complex, critical, multistep functions are correctly sequenced and ensures required steps are not overlooked.

If you use the orchestration component, Siemens Digital Industries Software recommends the following minimum sizing:

- **Minimum site consolidation clients**

The client compute platform should be sized to run the Teamcenter rich client (not required) and the client site consolidation interfaces. It should minimally be a 2 x 2 GHz processor with 4 GB of RAM and 120 GB of local storage.

- **Minimum site consolidation Teamcenter Integration Framework server**

The Teamcenter Integration Framework server must be able to run the Teamcenter Integration Framework application and mapping engine. It should minimally be sized at 4 GB RAM, 4 x 2 GHz processors, 120 GB of storage. The existing Teamcenter deployment infrastructure can be used for the database and other necessary Teamcenter components.

- **Minimum network connectivity requirement**

Network connectivity refers to client-to-server and inter-site connectivity. The internal data center server connectivity between Teamcenter tiers is assumed to be at least 1 GB and zero or nearly zero latency.

Minimum connectivity	Bandwidth	Latency
Consolidation client to server	LAN	< 5 msec
Intersite	45 MBit/sec	< 60 msec

- **Intermediate site file storage**

Storage for staging intermediate site consolidation files, such as database extracts, logs, and so on, is a function of the amount of information to be transferred concurrently.

- The source and target site exist in a common shared FMS network.

Performance monitoring

Consider the performance of the current infrastructure and the impact of site consolidation. In the execution phase, data replication can be run concurrently with normal production usage.

Monitor the following for signs of performance degradation:

- Clients
- All executing commands
- Enterprise servers
- Affected databases and storage

Client performance considerations

Memory usage, directory access, and log file tracking can be issues during tool execution.

- If you are using a Windows 32-bit, nonserver OS client, set the **/3GB** option in the Windows initialization (**boot.ini**) file to take advantage of the extended memory usage capability. On Linux, you can use 4 GB for the Teamcenter client to run the utilities.
- Executing site consolidation utilities requires access to the *TC_ROOT* and *TC_DATA* directories. This is part of the standard Teamcenter Foundation installation. A two-tier rich client located in the same local area network (LAN) as the database server is a suitable environment for performing the site consolidation activities. A four-tier rich client requires the Teamcenter Foundation software to be included in the setup for access to the utilities.

- Site consolidation utilities produce **log files** and output files that are important for tracking and monitoring the progress of the consolidation. The utilities and **readme** files provide information on the location and content of the log files.

Enterprise server tier issues

You must consider resources available to the Teamcenter enterprise tier server (pool server) during site consolidation activities. When you use Teamcenter Integration Framework orchestration, these resources include available **TcServer** processes and memory.

Teamcenter server processes are used with the orchestration portion of the installation. The memory resources for a single **tcservice** process can be very demanding. Memory usage ranges from 150 MB to 2.9 GB, depending on the operations being performed by the process. You must plan for additional resource usage, both in memory and CPU, during the site consolidation process. Review current usage of resources to ensure that there will not be limitations imposed on the site consolidation activities.

Database server considerations

Teamcenter site consolidation tools support consolidation activities for the following database and site types:

Database	Source site	Target site
Oracle	Supported	Supported
Microsoft SQL Server	Supported	Supported

See the Hardware and Software Certifications knowledge base article in Support Center for certified database versions.

Important considerations:

- The site consolidation utilities execute long-running SQL actions. You must size the memory cache appropriately to accommodate the additional demands on database processing. For Oracle, the System Global Area (SGA) and Program Global Area (PGA) must be sized to accommodate the current user load as well as the increased demands for site consolidation.
- Use your database vendor supplied tools to monitor your database and evaluate changes that may improve the database performance.
- You may need to create **table indexes** to accommodate increased demand on certain tables. As a first step, Siemens Digital Industries Software recommends you run the Teamcenter **index_verifier** utility against any source and target database to determine if there are any missing indexes.

Caution:

To manage potential hot disks containing the tables associated with Teamcenter, you can monitor the demand on the I/O subsystem supporting tablespaces. In particular, monitor the **POM_BACKPOINTER** and **PPOM_OBJECT** tables. These are very large tables that are traversed by the site consolidation utilities as well as your current user community.

Sites

Site definition data

Site definition data consists of administrative objects, class, and type definitions.

The target site must have definitions that are inclusive of those required by the data being transferred from the source site before that data can be properly imported. In some cases, defaults can be used, such as for groups or projects. In other cases, such as LOVs, you may need to modify values before you import them.

If you are using Teamcenter Integration Framework orchestration, the mapping can be performed in the mapping function of the groovy scripts initiated by the **sitcons_replicate_mgr** utility.

If you are using the low level functions of the **tcxml_export** and **tcxml_import** utilities to export and import the data, you must perform the necessary mapping on the data extracted from the source site prior to importing the data at the target site.

Administrative data

Administrative data, such as person, projects, groups roles, users, saved queries, and so on, can be exported and imported using the **admin_data_import** and **admin_data_export** utilities.

If the user that owns an object at the source site is not present at the target site and the **tcxml_export** and **tcxml_import** utilities are used to transfer a business object, the user that runs the **tcxml_import** utility becomes the owner at the target site. If you are using Teamcenter Integration Framework orchestration components to perform the import, the user that the Teamcenter Integration Framework connection uses to log on to the target site becomes the owner at the target site.

You may have other important administrative information that is not critical to the correct functioning of the site consolidation tools that you want to consider mirroring in the target site. This can include rule tree information, workflow templates, type-based subscriptions, and so forth. Create the administrative definitions at the target site using the specific tools and processes you would normally use to create and maintain them. For example, you can use Access Manager for rule tree information, Workflow Designer for exporting and importing work flows, and the Subscription Administration for type-based subscriptions.

You must manually create the list of users, groups, and disciplines to be notified using Schedule Manager notifications and subscriptions after importing the schedule at the target site.

Schema information

Schema information can generally be thought of as describing valid types, classes, relationships, and constraints, such as valid values, that represent the product data being managed.

- The target site must be prepared for site consolidation so the tools can import the consolidation product data into a correct and usable representation.
- The target site schema information must be prepared so the site functions as expected after the source site data becomes owned by the target site and the source site users become users of the target site.

You create and maintain this type of data using the Business Modeler IDE. You can use the **bmide_comparator**, **database_verify**, or the **bmide_commonplategenerator** utility to help you understand the schema differences between the databases. The **bmide_commonplategenerator** utility provides a human readable difference report.

Examples of the types of data include:

- Lists of values
- Tools
- BOM view types
- Dataset types
- Form types
- Occurrence note types
- Status types
- Item types
- Item revision types
- Relation types
- Any generic and user defined types, such as business rules
- Naming rules

2. Basic concepts for using site consolidation

Process definition assumptions

Major assumptions made in defining the site consolidation process include the following:

- Except for time critical periods, the source and target sites remain in production during the consolidation process.
- Consolidation is a binary operation. For example, a source site cannot be concurrently consolidated into more than one target site, and a target site cannot concurrently have more than one source site being consolidated into it.
- Consolidating a source site into a target site is done in an incremental process to support realistic performance goals and to give the user the flexibility to interrupt and resume the process.
- Analysis determines what sites are to be consolidated, and the source, target, and involved other sites are identified.
- The data, users, and process of the source site can be moved from the source to the target site in partitions.
- The data you are moving must be under the POM layer. If you require consolidation of data that is in a table that is not under the POM layer, such as the audit log history from Audit Manager, you must use database vendor's utilities to move the information in a separate process after the execution phase. For example, the audit log history is stored in the **USERDEFINEDLOG** and **AUDITLOG** tables. For an Oracle database, you can use the **exp** and **imp** Oracle utilities to dump the contents of these tables from the source site and import the tables into the target site.
- The functionality to move consolidation information during the execution phase requires you use only the source and target sites to extract, move, map, and load consolidation information.
- Network reliability, latency, and bandwidth are recognized as driving technical considerations for consolidation.
- Source and target sites:
 - Are in production operation on the same version of Teamcenter.
 - Can be made to have consistent site definition data, either through changes or metadata mapping.
 - Have compatible character representations at the database level.

- Have host clocks synchronized to the same GMT. If the host clocks are not close to synchronized, transfers may not update as expected as the last saved dates are used to determine out-of-date replicas.
- During the execution phase, there are no requirements for language translation of the following:
 - Metadata, such as LOV data.
 - Volume data, such as CAD files, Microsoft Word documents, and so on.

Note:

Siemens Digital Industries Software does not recommend moving the files in the Teamcenter volumes over the network at the same time as moving the Teamcenter database metadata. Therefore, Teamcenter provides a shared volume environment capability for use during and after the consolidation process. Siemens Digital Industries Software recommends that you disable the shared volume environment as soon as possible after completing the execution phase.

Site consolidation phases

Preparation phase

The preparation phase requires the following activities:

1. Deploy the additional Teamcenter components and the information technology (IT) environment required by the consolidation activity and the site consolidation tools. You must:
 - Install and configure the Teamcenter Integration Framework component if you are using it. The low-level TC XML site consolidation utilities are deployed when Teamcenter is installed.
 - Ensure the source and target sites are in a common, shared FMS network.
 - Ensure the accountability table is installed at the source site.
 - Ensure the site consolidation closure rules are loaded and the transfer options are set consistently at the source and target sites.

If there are local customizations to the data model, you may have to extend the closure rules so the instantiated **customized data gets extracted** and transferred.

2. Use the **Manage Administration Data** option in Teamcenter Environment Manager and the **generate_admin_data_compare_report** utility to ensure that the source site schema and administration data of the source and target sites are compatible. This is required to allow the target site to accept the product data from the source site.

3. Identify the product data to be consolidated using the **plm_report_extract** and **sitcons_gen_uidbatch** utilities. Structure the data into one or more partitions for each execution phase increment.
4. Use the Teamcenter rich client to locate collaboration contexts and move them to a folder. You can replicate this folder using either the **sitcons_replicate_mgr** utility or the low-level TC XML utilities.

The folder association during site consolidation transfers is maintained, so objects appear in the corresponding folders at the target site.

5. Use the following tools to identify data issues at both the source and target site that impacts the correct functioning of the site consolidation tools when moving the identified data:
 - Use the **review_volumes** utility to analyze and identify issues such as missing or extraneous files in the volume storage.
 - Use the **plm_report_extract** utility to identify issues and to suggest methodology and specific tools you can use to correct the issues.
 - You can correct issues using standard Teamcenter utilities, such as the **export_recovery**, **item_rename**, and **item_relink** utilities. For a list of utilities commonly used during site consolidation, see *Site consolidation utilities*.

Note:

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

Execution phase

In the execution phase, you move the identified product data in one or more increments from the source site to the target site. During each increment, you replicate one or more partitions of data and then reassign ownership from the source to the target site.

In this phase, volume references in the source data being imported are mapped by the site consolidation utilities to the appropriate shared volumes created at the target site during the preparation phase.

The typical structure of an execution phase increment consists of a replication period followed by the *time critical period* (TCP), in which final synchronization and ownership change are performed for the product data that was replicated. If you use the Teamcenter Integration Framework framework orchestration component to coordinate the operations performed during site consolidation, the primary utilities you use in the execution phase are:

- **sitcons_replicate_mgr**

Export, move, and import the data, and confirm to the source site that the target site has imported the data.

- **sitcons_user_folders**

Move user folders and/or update their content at the target site.

- **sitcons_xfer_owner_mgr**

Coordinate the change of ownership at the target and source sites.

If you do not use the Teamcenter Integration Framework orchestration component are responsible for performing the replication functions using these utilities:

- **tcxml_export** with the **-low_level** argument

Perform the initial export and subsequent synchronizing of updates.

- **tcxml_import** with the **-low_level** argument

Import the data exported by the **tcxml_export** utility. (The **tcxml_import -low_level** function uses **sitcons_replicate_mgr**.)

- **tcxml_confirm_export**

Confirm the target site import at the source site.

- **sitcons_user_folders**

Move the user folders and/or update their content at the target site.

- **tcxml_xfer_ownership**

Change ownership at the source and target sites.

You must also select a tool, such as FTP, to move data files to appropriate locations for processing by the site consolidation tools.

Cleanup phase

During the cleanup phase, you perform the activities that let you remove the source site from a Multi-Site federation and eventually decommissioned it if required. This includes the following activities at each site involved in the consolidation:

- Target site

Remove the source site volume references from the shared volume configuration and, if required, relocate the physical volume storage to the target site's data center.

- Other (third) sites

Update source site references to the data they contain.

- Object Directory Service (ODS)

Update references to source sites. In a multi-ODS environment, this operation must be done for all the ODS federations where the source site participates. If the source site participated in Multi-Site federations where the target site is not a member, the target site must be made a member so it can publish and share data within the other Multi-Site federations. Alternatively, you can remove the source site owned data or its ownership from the other Multi-Site federations.

A Multi-Site *federation* is an environment consisting of multiple Teamcenter sites known to each other, and sharing product data facilitated by Integrated Distributed Services Manager (IDSM) and ODS. Typically, sites in a federation sites have search, publish, or unpublish privileges for an ODS environment.

- Source site

Remove the site from the Multi-Site federation and, optionally, retire the site.

Use the `sitcons_fix_ixr`, `data_share`, and `data_sync` utilities to correct ODS and third site references to data from the source site that is now owned or shared with the target site.

Planning cleanup and volume relocation activities

Site consolidation presents options for performing cleanup activities and relocating data volumes.

- Perform cleanup activities between increments.

You can restructure the process to include the ODS cleanup and third site cleanup activities for a completed increment in parallel with other execution increments.

In general, the referenced file data is scattered unpredictably throughout a site's volumes. Unless you are performing the final execution increment and the source site is to be decommissioned, data that was replicated and had its ownership changed may still be referenced by objects not yet moved to the target site. Therefore, volume reassignment between increments is not usually possible.

- Determine whether volume relocation is required.

Whether to physically relocate volume storage to the target site is a business decision involving trade-offs between performance, reliability, and ease of administration for functions such as backup and recovery. This decision should reflect the improvement goals of the consolidation.

- To improve end-user performance when files are large, consider leaving the volumes near the end user. Moving volumes to locations that deliver these large files over WAN (wide area network) connections can negatively impact performance.
- To facilitate making backups and to centralize the data, consider moving the volume storage to the target site.

Teamcenter Integration Framework orchestration

Orchestration refers to the coordination of multistep operations performed during site consolidation.

You can use orchestration components to control the complex process of replication, syncing, and ownership transfer. Orchestration simplifies the user interaction, ensures all required steps are performed, and handles rollback on job failure.

Caution:

The Teamcenter Integration Framework supports and enables site consolidation orchestration. However, when Teamcenter Integration Framework is installed, Teamcenter users can transfer items using the rich client **Tools**→**Export**→**Remote using GMS** menu command. This can create problems if users attempt to transfer items during site consolidation. To avoid this possibility, **suppress the command**.

Two Teamcenter Integration Framework framework orchestration component utilities are provided to automate and control the more complex multistep operations of site consolidation:

- **sitcons_replicate_mgr**
- **sitcons_xfer_owner_mgr**

The script-driven processes are provided with Teamcenter, but can be modified using any text editor.

Tip:

The default time-out for processes is 10 hours. Consolidating site data usually requires large amounts of data to be transferred which can cause a process to exceed 10 hours to complete. If a transfer has stopped between process steps (seems stuck), increase the time-out value for the processes.

The **InvokeService.endpoint** file specifies the time-outs for the entire process. The **ProxyService.endpoint** file controls the time-outs for individual steps of a process sent through the proxy service. The time-outs are specified in milliseconds. For more information about configuring data exchange endpoints, see the **Apache ODE** documentation.

Groovy scripts automatically include the iForce mapping functionality. This mapping process allows types, lists of values (LOVs), and other values to be modified in transit to the target site. This adds time

to the replication process. Mapping can be eliminated by providing a null map or modifying the process to eliminate this step.

The following site consolidation operations are supported by orchestration utilities.

- Replication

Performs initial data movement as a replica from the source to the target site.

- Fast synchronization

Synchronizes the changes made at the source site with the already replicated data at the target site.

- Limited data verification

For the **sitcons_xfer_owner_mgr** and **tcxml_xfer_ownership** utilities, the **dryrun** argument outputs a report showing the objects that will have their ownership changed. When the **source_extinct** argument is used with the **dryrun** argument, the utility lists the data replicated to the target site from the source site by any other utility, and lists any changes to the initially imported data islands at the target site which are not present at the source site.

This validation can also be done using SQL, the **plm_report** utility, and a third-party tool, such as Microsoft Excel, to aid in comparison.

- Ownership change

Performs the ownership change at both the source and target sites.

For replication and ownership change, the orchestration process is as follows:

- Eliminates manual sequencing and operation of the tools underlying the replication and sync steps.
- Supports off-hours scheduling of tasks.
- Requires additional infrastructure because a middle tier is deployed.

Caution:

When monitoring orchestration, successive calls to get activity status in the Teamcenter Integration Framework user interface can cause JVM out-of-memory errors to occur. To avoid this, ensure that the application servers JVM memory setting is between 1200 and 1500 (optimally 1400). This is set by the application server instances JVM properties, for example **-Xmx1400m**.

- Places, on the transient volume directories, various outputs and log files of the utilities being operated by the orchestration framework.

- Collect these outputs for later troubleshooting analysis.
- If log files are configured to be retained, they must be cleaned up separately.
- Extract and status files generated by the underlying utilities are not automatically deleted by orchestration, so they must also be cleaned up separately.

The following example identifies the processes executed by the **sitcons_replicate_mgr** orchestration utility. It lists significant intermediate files created by the process. These files contain the next-step inputs, log files required by the process, and information useful for troubleshooting.

Process	Generated files
1. Scheduling	None
2. Data export	logfile10052009_092114.log trans4ac9f2ca0000478400005e99.log trans4ac9f2ca0000478400005e99.xml
3. Data mapping	1254748881752-56 1254748980986-58
4. Data import	logfile10052009_092437.log 1254748881752-5_importer.log 1254748881752-5_import_results.log 1254748881752-5_pretraverse.log
5. Undo data export	Called only in case of errors.
6. Confirm data export	1254748881752-5_import_results.txt low_level_file.txt

Process numbers reflect the default process activity order. The processes are similar for ownership transfer. The confirm ownership process (**-update_status**) is critical. It ensures data is owned by only one active site in the sharing environment.

When you use orchestration for replicating or ownership transfer, some of the steps may fail. It is your responsibility to monitor the processes and the **output logs** for failures. Some failures, such as a failure to connect or a connection time-out, are stored in the Teamcenter Integration Framework tables to allow the server to retry them at a later time. To prevent the failed steps from being attempted after you restart the server, you must remove them from the Teamcenter Integration Framework tables.

You can also set the **onerror.maximum.redeliveries** property to **0** in the **com.tc.esb.cfg** file to prevent the retry of the message and send it directly to the dead letter queue. You can manually move this message to the processing queue to retry the message.

Suppress the Teamcenter Integration Framework export command

To suppress the **Remote using Global Multi-Site** menu command, log on to the rich client as an administrator user and follow these steps:

1. Select the Command Suppression application.
2. In the **Organization** pane, under the **dba** group, select the **DBA** role.
3. In the **Application** pane, select **My Teamcenter**.
4. Expand **Tools** and **Export**, and select **Remote using Global Multi-Site**.
5. Click **Hide**.

The suppressed command is displayed with a strike-through line.

6. Click **Save**.
7. Repeat these steps for all user profiles.

Site consolidation tools and utilities

Teamcenter provides several utilities you can use to consolidate site data. The following table identifies helpful tools and utilities, what actions they provide, and when you use them.

Performance notes

Be aware that site consolidation utility operations mainly consist of traversing objects, resulting in lengthy execution times. For example, the **plm_report_extract** and the **tcxml_transfer_ownership -action=extract** utility operations are mainly object traversal.

The arguments of the utility can also cause different performance effects. For example, the **tcxml_transfer_ownership -action=perform** utility operation does not do object traversal, so it is not affected by extra database operations.

Teamcenter export operations experience a performance benefit from these extra database operations. Because Teamcenter export operations are performed more frequently than the site consolidation utilities are used, this results in better overall Teamcenter performance.

Site consolidation utilities

Tool or utility	Action	Phase
<code>admin_data_export</code> and <code>admin_data_import</code>	Distribute system administration data, such as users and groups, from one site to another. When adding a new site, the utilities let you enter the site information for all sites in the network so the new site can exchange data with them.	Preparation
<code>Business Modeler IDE</code>	Allows analysis, comparison and configuration of data models. For example, you can compare two complete Teamcenter model files and generate a differences file.	Preparation
<code>database_verify</code>	Compares database schema, Teamcenter types, tools, release statuses, and units of measure between two specified Multi-Site Collaboration sites and generates a report of any database discrepancies.	Preparation
<code>cleanup_shared_objects</code>	Use to clean up ownership inconsistencies at all sites for a given item and its dependent objects. The focus of this utility is on VEBs, notes, or alternates.	Preparation and cleanup
<code>data_share</code>	Performs various Multi-Site Collaboration operations, such as publishing and unpublishing objects collectively and sending objects to remote sites. It can be used as a deployment tool during the initial Multi-Site Collaboration implementation phase or as a daily tool to perform actions like publish and unpublish objects, list ODS sites, and delete ODS publication records.	Preparation, execution, and cleanup
<code>dataset_cleanup</code>	Repairs corrupted datasets and removes orphaned revision anchors.	Preparation
<code>data_sync</code>	Synchronizes copies of objects at remote sites with the latest version of the primary object. It also updates publication records when republishing objects. In verify mode, the utility checks the existence of exported objects at the remote sites; if a copy no longer exists at the remote site, the corresponding import export record is deleted from the owning site.	Preparation, execution, and cleanup
<code>ensure_site_consistency</code>	Performs corrective actions required when a synchronous site transfer (SST) ownership transaction is interrupted due to a system or network crash or a user-initiated process termination.	Preparation and cleanup
<code>export_recovery</code>	Recovers and restores exported objects to your database under certain conditions. Imports multiple items (in batch mode) into the Teamcenter database. It is the companion to the <code>item_export</code> utility.	Preparation and cleanup
<code>generate_admin_data_compare_report</code>	Provides an HTML formatted report that shows the differences between the target and source sites.	Preparation
<code>import_file</code>	Imports files into the Teamcenter database according to a set of user-specified arguments. These arguments supply user identification information, dataset information, and (optionally) item information to be associated with the imported file.	Preparation and cleanup

Tool or utility	Action	Phase
item_export	Exports a single item or multiple items in batch mode. It is the companion to the item_import utility.	Preparation and cleanup
item_import	Imports multiple items (in batch mode) into the Teamcenter database. It is the companion to the item_export utility. This utility supports part family templates and members.	Preparation and cleanup
item_relink	Allows you to replace the external references for a duplicate item and its corresponding replica. It is the companion to the item_rename utility.	Preparation and cleanup
item_rename	Allows you to correct naming conflicts. Companion to the item_relink utility.	Preparation and cleanup
plm_report_extract	Extracts persistent data from the Teamcenter database and store it in intermediate binary (not human readable) format. The extract is run independently for each site.	Preparation
plm_report_consistency_analysis	Performs consistency analysis based on the set of extracts input to it, and also provides capability to generate a text file from an extract that is readable or can be input to other applications such as Microsoft Excel. This utility is a companion to the plm_report_extract utility.	Preparation
plm_report_constraint_analysis	Analyzes the target database and reports potential database constraint violations, such as duplicate item IDs. This utility uses the output files generated by the plm_report_extract utility.	Preparation
purge_datasets	<p>Removes (purges) old versions of datasets from the database. Normally, Teamcenter stores a fixed number of dataset versions in the database. The maximum number of datasets retained is set using the AE_dataset_default_keep_limit preference. However, certain conditions can prevent automatic purging of old datasets, for example when:</p> <ul style="list-style-type: none"> • A user does not have permission to purge a dataset owned by another user. • A group is given read/write permission but not delete permission. <p>The utility produces a listing that shows each dataset purged, along with the owning user and group.</p>	Preparation and cleanup
review_volumes	Allows viewing detailed information about Teamcenter volumes and removal of unreferenced operating system files from these volumes.	Preparation and execution
sitcons_accountability_chkd	<p>Generates a file listing the objects in the database that are not part of the consolidated objects.</p> <p>The SITCONS_AUTH_KEY environment variable must be set to run this utility.</p>	Preparation
sitcons_extract_shared_vols	Generates a file used to create the shared volume for the source site at the target site.	Preparation

Tool or utility	Action	Phase
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
sitcons_fix_ixr	Updates export records at sites that have replicas or primary versions of consolidated data to reflect the correct site replication and ownership.	Cleanup
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
sitcons_gen_shared_vols	Creates shared volume definitions at the target site, updates the primary FMS, and creates volume mapping for data consolidated at the target site.	Preparation
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
sitcons_gen_uidbatch	Generate a set of files containing a number of UIDs of a given class.	Preparation
sitcons_replicate_mgr	Performs fast data replication of source data using Teamcenter Integration Framework. Supports parallel export/import tasks.	Execution
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
sitcons_user_folders	Transfers a user's Home , Newstuff , and Mailbox folders from the source to the target site.	Execution
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
sitcons_xfer_owner_mgr	Transfers ownership of low-level consolidated objects from the source database to the target database using the Teamcenter Integration Framework orchestration process.	Execution
	The SITCONS_AUTH_KEY environment variable must be set to run this utility.	
tcxml_confirm_export	Updates the source site accountability tables with the results of the target site import.	Execution
tcxml_export	Exports bulk consolidated data in Teamcenter XML (TC XML) format. This utility is a companion to the tcxml_import utility.	Execution
tcxml_import	Imports Teamcenter site data from the Teamcenter XML (TC XML) format. This utility is a companion to the tcxml_export utility.	Execution
tcxml_xfer_ownership	Changes ownership at a site. It is used by the sitcon_xfer_owner_mgr utility orchestration.	Execution

3. Preparing for site consolidation

Deploying Teamcenter components

Deploying the additional Teamcenter components and IT environment elements required for consolidation by the site consolidation tools is a required initial step. Subsequent activities during the preparation and execution phases depend on this.

Deployment includes the following activities:

- Upgrading the target site for any components required by the data being consolidated that do not already exist.

For example, if the source site has an NX Integration and the target site does not, the target site must be upgraded to the same version or you cannot reliably analyze and move the source data.

When the target site deployment has everything the data being imported from the source site needs, this task is done.

- Verifying the accountability tables are installed at the source site. These tables are used to track the data being consolidated and its state during the consolidation process, such as exported, imported, modified since initially replicated, up-to-date at the target site, ownership changed, and so forth.

Set the **TC_TIE_TS_NAME** environment variable to the tablespace you want to use for the site consolidation tables. Create the tables using the **install** utility with the **-tie_sitecon_support** and **-tie_configbom_support** arguments.

Install the table triggers, using the **install** utility with the **-tie_triggers -u -p -g** arguments. Doing so installs the **fast_sysnc_add_trigger** and **fast_sync_delete_trigger** triggers to support data synchronization.

Note:

When creating the accountability tables the trigger creation may fail without any descriptive error. To identify the problem or failure point, use the following command.

```
SHOW ERROR TRIGGER trigger_name
```

- Verifying the closure rules and transfer options are correct and the same at the source and target sites. These rules and options ensure the traversals find the appropriate data. They are key components of fast import and export. Extension of the closure rules may be necessary if you have customizations.

- Creating the environment between the source and target sites so the volume storage of the source site is shared and the source site files do not have to be moved during the execution phase. The environment must have:
 - Source and target sites in the same shared FMS network. The shared FMS network must be configured before you perform the steps to create the shared volume environment.
 - Shared volume definitions in the target site of all source site volumes, an updated FMS primary configuration file, and a preference map that indicates how the replication tools map volume references in the source site metadata when it is imported into the target site.

The high-level steps required to accomplish these tasks are:

1. Establish a shared FMS network between the source and target site.
2. Validate the shared FMS network configuration.
3. Extract the shared volumes from the source site using the **sitcons_extract_shared_vols** utility.
4. Generate the shared volumes at the target site, the updated FMS primary configuration file, and volume preference map using the **sitcons_gen_shared_vols** utility.

Note:

The **FSC_DelayedVolumeValidation** FSC configuration parameter can be used to improve the performance of the **sitcons_gen_shared_vols** utility.

5. Validate the **FMS network shared volumes** configuration.

Upgrade the ACCT_TABLE database table

Use one of the following manual methods to complete the upgrade of the **ACCT_TABLE** database table:

- To update table for site consolidation unconfigured synchronization, use **:SITECON_SYNC** as the **app_id** argument, for example:

```
tcxml_acct_table_upgrade -u=admin-user-id -p=password -g=dba -app_id=:SITECON_SYNC
```

Then do full object data re-indexing to regenerate data with the proper **app_id** value.

- To duplicate all records with no data loss but some unnecessary records, use **:SITECON_SYNC,:FTS_SYNC** as the **app_id** argument, for example:

```
tcxml_acct_table_upgrade -u=admin-user-id -p=password -g=dba  
-app_id=:SITECON_SYNC:FTS_SYNC
```

- To update the table for a specific use case, use either **:FTS_SYNC** (for full-text search) or **:SITECON_SYNC** (for site consolidation unconfigured synchronization) as the **app_id** argument, for example:

```
tcxml_acct_table_upgrade -u=admin-user-id -p=password -g=dba -app_id=:FTS_SYNC
```

or

```
tcxml_acct_table_upgrade -u=admin-user-id -p=password -g=dba -app_id=:SITECON_SYNC
```

- To clean up all records in the table and rerun the use cases to regenerate data with the proper **app_id** value, use the **-delete** argument, for example:

```
tcxml_acct_table_upgrade -u=admin-user-id -p=password -g=dba -delete
```

Once you have completed the upgrade of the **ACCT_TABLE** database table, use the following SQL command to remove all records from the table having **app_id =:SITECON_SYNC**.

```
delete from acct_table where app_id =':SITECON_SYNC'
```

Maintain data synchronization through site consolidation

All Teamcenter applications have an application ID. All applications that use TC XML and perform data synchronization have their application IDs registered in the Teamcenter database's subscription table (**SUBSCRIPTION_TABLE**). If an application ID is inactive longer than the time specified by the **install** utility's **TC_TIMESTAMP_THRESHOLD** parameter, it is deleted from the subscription table. When this happens, the related data is no longer synchronized.

Set **TC_TIMESTAMP_THRESHOLD** appropriately for site consolidation and system upgrades

By default, **TC_TIMESTAMP_THRESHOLD** is set to a value of **96** hours.

- Consider increasing this value when performing a site consolidation (and system upgrades) that may take several days to complete. Doing so allows data synchronization to continue after the consolidation or upgrade completes.
- High values are not always best. **TC_TIMESTAMP_THRESHOLD** also controls the interval for stale data cleanup. Timely data cleanup results in better data synchronization performance. If you increase the value for a site consolidation or system upgrade, reduce the value after the consolidation or upgrade completes.

Manage synchronization notifications

By default, a notification email is sent to the current user 24 hours before an application ID is to be deleted from the subscription table and again when the application ID is deleted from the subscription table. The user can then choose to rerun the use case or rerun the export or import.

To limit email notifications, set the preference **Subscription_Table_Notify_Level** as follows:

- 1 Send an email notification only when an application ID is deleted from the subscription table.
- 2 Send no email notifications.

Unset **Subscription_Table_Notify_Level** to return to the default notification.

To specify additional email notification recipients, set **Subscription_Table_Notify_Users** to the Teamcenter user IDs or external email addresses, separating multiple values with commas.

Making site definitions compatible

Rationalizing site definitions

Site definition rationalization makes the target site and the source site administrative and schema data compatible so the target site can accept the product data that is to be moved from the source site.

The simplest approach is to ensure the target site definition data (organizational and schema information) is a superset of the of the source and target site. The major analysis tools provided include the **admin_data_export** utility, **admin_data_import** utility, **database_verify** utility, the Business Modeler IDE, and Access Manager from an actual alignment perspective (for example, schema alignment and organizational data movement).

Access Manager is less critical to the site consolidation tools because the tools must be run by a user with Teamcenter administration privileges. However, it is critical to ensure correct system functioning in the post consolidation production operation period.

Consolidating administration data

Teamcenter Environment Manager (TEM) allows you to export and import organization data provided the two sites are both at the current version. The **generate_admin_data_compare_report** utility provides a report that shows the differences between the two sites.

Site consolidation preparation involves consolidation and transfer of the administrative data from the source site to the target site. This is required when the users and the business process of the source site must be transferred to the target site after the data transfer.

The current functional behavior of this utility supports reporting and transferring of **Person**, **User**, **Group**, **GroupMember**, **Role**, **POM_Inc**, **Projects**, and **IMANQuery** classes of the Organization application. The following process describes how to consolidate and transfer the administrative and organizational data:

- Export the Organization application objects at the source site to an XML file using TEM.

- Identify those corresponding object instances to be transferred to the target site required by the transfer of business processes.
- Export the instances of the organization data on the target site as an XML file using TEM.
- Use the **generate_admin_data_compare_report** utility to identify differences in the Organization at the two sites. Use the report to identify and resolve any conflicts in the Organization application objects.
- Use TEM to import the organization objects from source to target site required for the business process of the users at both the source and target site.

Process for consolidating administrative data

The best practice is to use the **Manage Administration Data** feature of Teamcenter Environment Manager (TEM) and the **generate_admin_data_compare_report** utility for mass consolidation of administrative data when both sites are at the current version of Teamcenter. For small sites with few users, groups, roles, and so on, it may be equally appropriate to do this manually.

1. Perform a comparative analysis to understand how to reconcile the administrative data of the source and target site.
2. After you complete the analysis and determine how to address the issues, make separate export runs for each supported class:

```
pom_inc
role
person
group
user
groupmember
savedqueries
projects
```

3. Make the changes determined in step 1 to avoid duplicates. Append a unique suffix to each person name in the source site export file to avoid conflicts of unique duplicate person objects.
4. Prevent the import of **ApplicationRef** elements by performing one of the following:
 - Manually delete all **<ApplicationRef...> ... </ApplicationRef>** entries from the export files.
 - Use a closure rule that ignores all **ApplicationRef** elements during import as follows:
 - a. Create a new transfer mode using a copy of the **incremental_import** transfer mode and the closure rule associated with it.
 - b. Open the closure rule and add a new clause as follows at the top of the clause list:

Primary object class type	Primary object	Secondary object class type	Secondary object	Relation type	Related property or object	Action type	Conditional clause
TYPE	PIESession	CLASS	ApplicationRef	CONTENT	*	SKIP	

Or manually edit the closure rule by adding the following at the top:

```
TYPE PIESession CLASS ApplicationRef CONTENT * SKIP
```

5. Import the modified source export files into the target site. This adds new objects and overwrites existing duplicate objects so the analysis of the source export files can eliminate unwanted duplicates. Siemens Digital Industries Software recommended that the classes be imported in the following order:
 - a. **pom_inc**
 - b. **role**
 - c. **person**
 - d. **group**
 - e. **user**
 - f. **groupmember**
 - g. **savedqueries**

Rule trees

Rationalizing Access Manager rule trees

Access Manager allows administrators to implement a corporate data access strategy by configuring rules. Access Manager allows an administrator to create, update, or delete rule tree nodes and named access control lists (ACLs).

Rule tree rationalization is the consolidation of the source and target site rule trees into a rule tree to be used by Teamcenter to compute the access privileges for requests to the target site for data managed by the target site.

Access Manager rule trees from source and target sites can be rationalized one of two ways:

- Simply, by adding a node in the target site rule tree hierarchy that represents the source site rule tree configuration.

- At a more granular level, by aligning roles and access privileges from a business process perspective.

The more granular approach requires more analysis, iteration, and review, prior to consolidation, to ensure business processes with both source and target data continue without impact.

If you modify an object's ACL at the source site you must manually modify the object's ACL at the target site after the object is transferred either manually or by calling the following Access Manager APIs programmatically through a customization:

- **AM_grant_privilege**
- **AM_revoke_privilege**
- **AM_unset_privilege**

A modification to an object's ACL cannot be transferred using export utilities.

Following are some best practices you should consider when integrating Access Manager rule tree structures between source and target sites:

- **Do not modify ACLs referenced by rules on the System Objects branch of the rule tree.**

Adding new rules, deleting rules, or in any way modifying existing rules on the **Systems Objects** branch of the rule tree may result in unpredictable behavior or loss of data.

- **Do not modify the upper area of the rule tree.**

Deleting branches or changing the order of branches in the upper area of the rule tree must be done with great care because these changes have significant impact on the evaluation precedence of rules. Modifications in the upper area of the rule tree may result in unpredictable behavior.

- **Do not use a text editor to modify rule tree files.**

Rule tree files are simple ASCII files and conform to a particular format. You can read rule tree files using any text editor; however, modifying them with a text editor can easily corrupt the file.

- **Populate access control lists (ACLs) sparingly.**

Explicitly grant privileges, and only deny privileges when users must be blocked from privileges that would otherwise be implicitly granted.

- **Whenever possible, leave privileges unset.**

Leaving privileges unset in ACLs allows rules to accomplish focused objectives, and it also allows objects and accessors to filter through rules that do not apply to them.

- **Define relevant ACL names** ACL names are displayed in the rule tree and in dialog boxes throughout the Teamcenter interface.

You can significantly enhance overall usability by defining these names carefully. For example, when creating an ACL for working data, name it according to the data type (for example, item, item revision, or **UGMASTER** rather than a role name or some other description).

- **Understand your organization's business rules.**

A thorough understanding of your organization's business rules enables you to model access rules that support your business processes and are transparent to users. When modeled correctly, Access Manager rules grant users the privileges necessary to perform the tasks associated with their jobs while denying them access privileges to data that is released or out of the scope of their functional role.

- **Document the business rules and the rule tree developed to meet them.**

Every rule in the rule tree and the associated Named ACLs are included for a purpose. For maintenance purposes, Siemens Digital Industries Software strongly recommends to document the purpose of the rules, how they are populated, and why they have been populated. Future versions of Teamcenter add new rules and accessors. Merging new rules and accessors is a manual process, which is simplified if the Access Manager rule tree is thoroughly documented.

- **Add new rules for working data in the Working data branch of the tree.**

The proper location to add new rules for working data is under the **Working** data branch in the rule tree. This helps to customize the rule tree and identify working data.

- **Use the Has Attribute rule to create custom rules based on any attribute of an object of a given class.**

For example, if you have the following rule:

```
WorkspaceObject:object_name=*x PublicationRecord:security=suppliers
```

The class and attribute names are not case sensitive. The attribute type can be string, double, integer, logical, or reference. This rule supports custom attributes.

- **Use discretion in applying the Bypass ACL.**

The **Bypass** ACL grants all privileges to system administrators who have set **User Status Bypass** to **ON**. Use discretion in applying this ACL.

Access Manager tasks

You can use Access Manager to accomplish the following tasks:

- Create, modify, and delete a rule tree node.
- Create, modify, and delete a named ACL.
- Import a rule tree from a file.
- Export a rule tree into a file.
- Move a rule tree node up or down, so that the rule hierarchy and precedence changes.
- Cut and paste the rule tree node under a different parent.

Access Manager objects

The main object in Access Manager is **AM_tree**, which represents the Access Manager rule tree node (which is a child tree).

- The **AM_ACL** object represents the named ACL object.
 - An Access Manager tree node has an association with access control lists (ACLs).
 - The Access Manager tree node can be associated with zero or one ACLs.
 - One ACL object can be associated with zero or more Access Manager tree nodes.
- The **AM_ACE** object represents the access control entry (ACE) object in an ACL.
 - An ACL can have one or more ACE objects.
 - An ACE object can be associated with zero or more ACL objects.
- The **AM_named_tag** object represents the direct accessor type objects.
 - An ACE object can be associated with one accessor type.
 - An accessor type can be associated with zero or more ACE objects.
- The **AM_privilege** object represents the privilege object.
 - Each ACE can be associated with zero or more privileges.
 - Each privilege can be associated with zero or more ACE objects.

Reconciling rule trees

Avoiding common mistakes

Rule tree reconciliation can be a difficult task if each rule and position is not completely understood. This includes condition and value placement, ACL accessors, and their rights.

You must understand:

- The site core data model, driving use cases, and general access orientation based on organization.
- How to read an exported rule tree.

To avoid causing problems during reconciliation:

- Do not modify the upper area of the rule tree.
- Do not use a text editor to modify rule tree files.
- Populate access control lists (ACLs) sparingly.
- Do not modify ACLs referenced by rules on the system objects branch of the rule tree.
- Define relevant ACL names.

ACL names are displayed in the rule tree and in dialog boxes throughout the Teamcenter interface.

Prerequisites for rule tree reconciliation

Reconcile rule trees before you import workflows. Before reconciling rule trees, perform the following tasks:

- Reconcile data models.
- Reconcile organization objects.
- Reconcile project objects.

In most cases, two active sites must be consolidated, and you must still understand the difference between default behavior and customized behavior.

For example, a new target site rule tree is different from the default rule tree, and a source site rule tree is also different from the default rule tree. Compare the differences at target site to the differences at the source site to determine the actions needed to correct discrepancies.

Reconcile rule trees

1. Obtain a default rule tree. This becomes the new base (primary) rule tree. You should use a copy of a default rule tree from the new target Teamcenter version.
 - a. Log on to the target site where the rule tree has not been modified.
 - b. Open the Access Manager application.
 - c. Choose **File→Export**.
 - d. Name the file **default_rule_tree.txt** and click **Export**.
2. Obtain the rule tree from the site to be consolidated.
 - a. Log on to the source site.
 - b. Open the Access Manager application.
 - c. Choose **File→Export**.
 - d. Name the file *site_name_rule_tree.txt* and click **Export**.

Note:

Exported file characteristics may be different based on the database born-on date. Do not overlook the next few steps.

3. Compare the **default_rule_tree.txt** content to the *site_name_rule_tree.txt* content.

The three key sections in every exported rule tree file are header, ACLs, and condition value placement.

4. Validate that the header is identical for the two files. For example:

If the **default_rule_tree.txt** file has:

```
READ!WRITE!DELETE!CHANGE!PROMOTE!DEMOTE!COPY!CHANGE_OWNER!PUBLISH!
SUBSCRIBE!EXPORT!IMPORT!TRANSFER_OUT!TRANSFER_IN!WRITE_ICOS!ASSIGN_TO_PROJECT!
REMOVE_FROM_PROJECT!REMOTE_CICO!UNMANAGE!IP_ADMIN!ITAR_ADMIN!CICO!
```

And if the *site_name_rule_tree.txt* file has:

```
READ!WRITE!DELETE!CHANGE!PROMOTE!DEMOTE!COPY!CHANGE_OWNER!PUBLISH!
SUBSCRIBE!EXPORT!IMPORT!TRANSFER_OUT!TRANSFER_IN!WRITE_ICOS!ASSIGN_TO_PROJECT!
REMOVE_FROM_PROJECT!REMOTE_CICO!
```



```

Has Class( POM_object )->System Objects , Collapsed , Protected
  Owing User( infodba )->System
    Has Class( RevisionRule )->Public Rev Rule
    Has Class( some_custom_object )-
>infodba_owned_some_custom_object
  Has Name( Mailbox )->Mailbox
  Has Class( ImanAliasList )->Personal Address List
  Has Class( Folder )

```

There are several additional objects in the new default rule tree. This can be a result of the creation date or administrator modification.

- Generally, you keep the listings in the new default rule tree and you do not push any of the old rules from the *site_name_rule* tree. This is because most new entries are simple strategy changes and additions through evolution of the data model.
 - When there has been a justified or calculated change for functional purposes, such as an addition to the out of the box rule tree, you may be required to propagate a change to the new default rule tree file. For example, **Has Class(some_custom_object)** represents this type of exception.
6. Add missing conditions, values, and rules to the new rule tree. This is an iterative process you must apply to the entire rule tree.

Typically, it is a best practice to not modify the rule tree above **Has Class(POM_application_object)**. However, when this is required, you have two options:

- Reposition the rule under **POM_application_object** in an attempt to follow the best practice.
- Propagate the change into the new primary rule tree.

Move the condition and value to the new rule tree. To do this, create the condition and value using the rich client or modify the rule tree text file.

When a condition and value are moved to the new tree, take care to follow the format of the file precisely. If you use tabs or poor spacing, you will encounter issues on import.

Regardless of condition and value placement, you must move the **custom_objects_acl_some_custom_object** rule to the new file. To do this, you can create the rules using the rich client or modify the rule tree text file.

- To modify the text file, find the **custom_objects_acl_some_custom_object** rule in the ACL section of *site_name_file*.

For example:

```
custom_objects_acl_some_custom_object!RULETREE
Owning User!! !Y!Y!Y! !! !Y! !! !! !! !! !! !! !! !! !! !! !! !! !!
World!! !N!N! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !!
```

- When you find this rule, make sure that the type and ID of the accessor exists at the target location.

Caution:

Rules impact functionality; ensure this fits your use case.

- After the file is found, add the copied rule to the bottom of the new file ACL section. Ensure proper document form, including trailing spaces and single lines between rules.

For example:

```
Invisible Program!RULETREE
World!!N! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !!
Has Class( POM_object )
  Has Bypass( true )->Bypass
  Has Class( POM_object )->System Objects , Collapsed , Protected
  Has Class( WorkspaceObject )
    Inactive Sequence( true )->Inactive Sequence Objects
  Is Archived( true )->Archived Objects
```

Becomes:

```
Invisible Program!RULETREE
World!!N! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !!
custom_objects_acl_some_custom_object!RULETREE
Owning User!! !Y!Y!Y! !! !Y! !! !! !! !! !! !! !! !! !! !! !! !!
World!! !N!N! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !! !!
Has Class( POM_object )
  Has Bypass( true )->Bypass
  Has Class( POM_object )->System Objects , Collapsed , Protected
  Has Class( WorkspaceObject )
    Inactive Sequence( true )->Inactive Sequence Objects
  Is Archived( true )->Archived Objects
```

7. Manage multiple-site issues.

When you consolidate rule trees, the same condition and value can exist between the locations, but may contain different rules.

- A rule can exist at two sites, but be different at the two locations.

In this situation the business and use case must to be understood and a choice must be made.

- A condition and value can be the same yet use completely different rule.

Both ACLs may have the same intent, so you can pick one method and use it, but if the ACLs have different intents, you must decide which rule to use.

8. Examine workflow ACLs.

The rule tree contains **WORKFLOW** ACLs. These are created in the Workflow Designer application and exist in the ACL section of the rule tree export file.

If you import workflows before this step, the rules must be reattached to their task.

- Identify all **!WORKFLOW** entries in the rule tree.

```
During_a_workflow_task!WORKFLOW
Role!CHECKER!Y!Y!N!N!N!N!N! ! ! ! ! ! ! ! ! ! ! !
Group!Drafting!Y!N!N!N!N!N!Y! ! ! ! ! ! ! ! ! ! ! !
```

- Copy the ACLs to the new rule tree.

9. Examine each part of the rule tree.

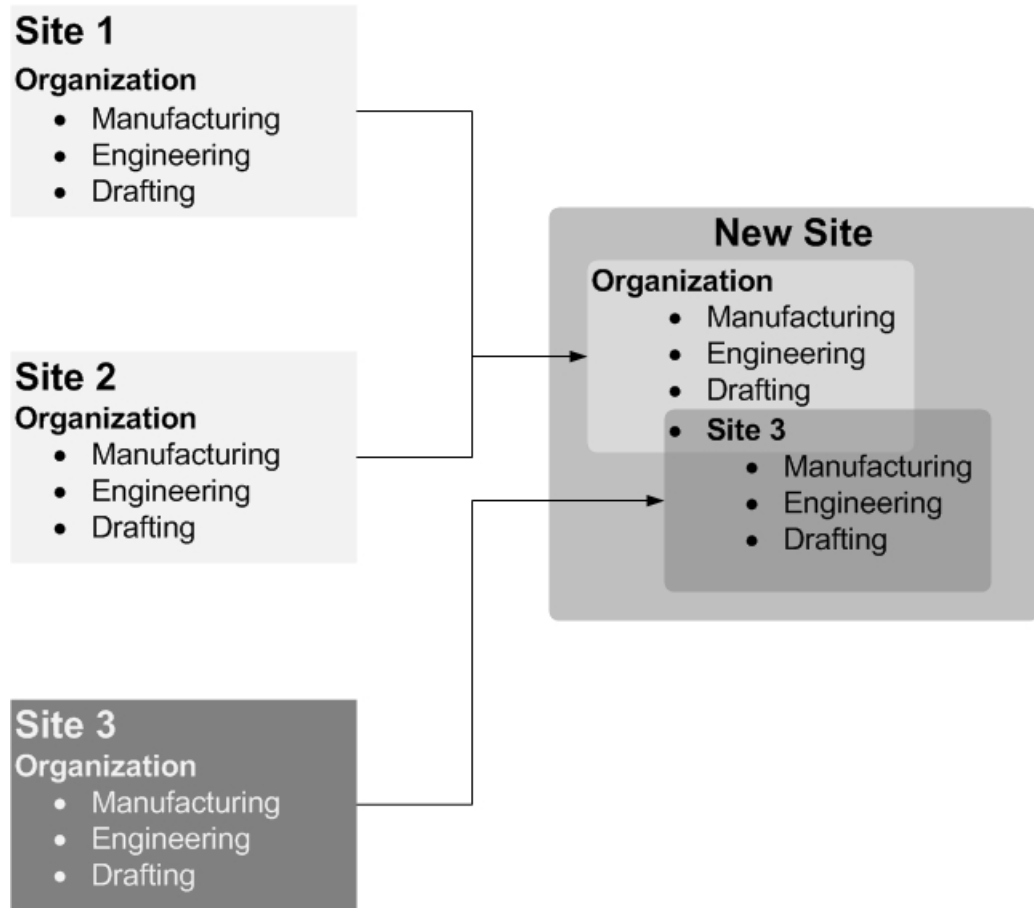
You must understand what exists in the out-of-the-box ACL and identify what is changed at each respective location.

10. Use a subgroup access method.

When a consolidation occurs, some sites may need to retain their own owning-site access rules, even after being integrated to the new site. There are several ways allow the once independent site to retain integrity.

Consider the situation of consolidating three sites.

- Organizations from site 1 and site 2 are to be integrated into the existing groups at the new target site.
- Site 3 must be repositioned to either a newly created group at the same level, or one level down, to support workflow assignments by group and role, group access needs, or to minimize confusion in the user community.



For example, under the **Has class (POM_application_object)->Working** line, you can add a new rule.

```

Has Class( POM_application_object )->Working
  Has Type( NXDerived )->NXDerived Access
  Is GA( true )->GA Working
  
```

Becomes:

```

Has Class( POM_application_object )->Working
  Owning Group( *.site3 )->Site3_rules
  Has Type( NXDerived )->NXDerived Access
  Is GA( true )->GA Working
  
```

- You can create an ACL that provides access to all subgroups of the site 3 group, already attached. In this case, the rule closely resembles the working ACL.

```

Site3_rules!RULETREE
Group!Site3!Y!Y!Y! ! ! !Y!Y!Y! ! ! ! ! ! ! ! ! ! ! ! !
Group!Drafting.Site3!Y!Y!Y! ! ! !Y!Y!Y! ! ! ! ! ! ! ! ! ! ! !
Group!Engineering.Site3!Y!Y!Y! ! ! !Y!Y!Y! ! ! ! ! ! ! ! ! ! ! !
  
```

```
Group!Manufacturing.Site3!Y!Y!Y! ! ! !Y!Y!Y! ! ! ! ! ! ! ! ! ! !
World!!N! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
```

- You can add nested ACLs as required. For example, to share any standard part that is owned by site 3, you can apply the following rule:

```
Has Class( POM_application_object )->Working
  Owing Group( *.site3 )->Site3_rules
    Has Type( Standardpart )->Site3_owned_standardpart
    Site3_owned_standardpart!RULETREE
    World!!Y! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
```

To optimize performance, minimize the use of nested rules.

- Use the Teamcenter command shell to import the changes to the rule tree file.

```
am_install_tree.exe -u=tadmin -p=password -g=group
-path="C:\work\ACLs\Default_rule_tree.txt" -mode=replace_tree
```

Consider importing the text file after every few changes to validate that there are no syntax errors.

Detecting and correcting data issues at the source and target sites

Use the Business Modeler IDE to make other administrative and schema data such as LOVs, types, rules, and extensions compatible between the sites.

The Business Modeler IDE is a tool for creating data model extensions and configuring the business behavior of Teamcenter. Teamcenter is easily configurable through both *codeless* and *codeful* extension points.

Codeless extension points are places within the Teamcenter code base that expose a configuration point through a graphical user interface. Wizards are used to guide the user through configuring the extension point to tailor Teamcenter to align within a business process. Some examples of codeless extensions are business objects and properties, naming pattern validation, number sequence generators, deep copy rules for revision, relation cardinality, and pick lists for properties.

Codeful extension points are places within the Teamcenter code base where code can be authored to provide advanced business behavior. The Business Modeler IDE provides the mechanisms that enable the processes needed for creating, organizing, managing, and distributing extensions through a template.

- Business objects are the primary objects that Teamcenter users work with to create product data.

These business objects are modeled in the Business Modeler IDE. Business objects contain properties and behaviors and are hierarchical in nature. A new business object can be added anywhere in the hierarchy to inherit behavior from its parent business object, such as:

Item
Form
Dataset
AppInterface
GDELink
IntermediateDataCapture
StructureContext

- Properties are the attributes that describe the business object. Consider the following:

Persistent
Compound
Run time
Relation properties

- Your schema's classes and attributes are the logical database representation of the business objects and properties.
- Business rules are the business behavior extension points of Teamcenter. The following all have rule attributes as an additional extension point:

Display
Naming
Extension
Deep copy
GRM
ID context
Property
Compound property

- Lists of values (LOVs) provide specific values for a given property in the user interface to when these values must be:

Standard
Filtered
Hierarchical
Interdependent

- Other extension points are available for configuring tools, change, unit of measure, and so on, such as:

Change
Tool
Status
View
PS occurrence
Validation
ID context

Unit of measure
Storage

- A new constants framework is available by which you can configure extension points of the system and provide some of your own. These include:

Global
Business object
Property constants

- A rules engine framework is available for providing robust decision table logic for configuring system behavior. This provides application extension points and rules and business context.

Schema compatibility

Identifying source and target schema differences

The **database_verify** and **bmide_comparator** utilities produce change files that list the differences between the source model and the target model. For testing, always set up the old model path to be the expected XML definitions from the templates and the new model path to be the extracted model from the database. If the change file has zero elements in it, there are no differences between the source and target models. If there are differences, they must be analyzed. The XML files are very straightforward and easy to read. The change file is divided into three main sections:

- <Add>** These elements contain schema information from the source site that does not exist at the target site.
- <Change>** These elements indicate a difference between the sites for the same element.
- <Delete>** These elements indicate schema information that does not exist at the source site. It can usually be ignored during site consolidation.

Depending on the observations made from the analysis phase of the schema, add, delete, and modification activities may be needed.

Resolving Add element issues

Any element that appears in the **Add** element exists in the old model but not in the new model. It means that the **Add** elements are in the extracted database model but not in the expected template XML model. Therefore, there are utilities/commands run as a part of installation/upgrade that are adding elements that are not reflected in the XML template. Either remove the commands or add the elements to the template.

For example, if the change file has the following **Add** element:

```
<Add> <TcStandardType typeName="TraceLink"
      parentTypeName="ManagedRelation" typeClassName="TraceLink"/>
```

```
</Add>
```

A **TraceLink** type is added to the database but is not listed in the template XML.

Resolving Delete element issues

Any element that appears in the **Delete** element does not exist in the new model but does exist in the old model. Therefore, the **Delete** elements are not in the extracted database model but are in the expected template XML model. There are utilities/commands that are missing in the installation/upgrade scripts that should be adding the elements. Either add the commands or remove the definitions from the template. For example, suppose the change file has the following **Delete** elements:

```
<Delete> <TcAttributeAttach className="CAEAnalysisRevMasterStore">
  <TcAttribute attributeName="solution_step" attributeType="POM_string"
  maxStringLength="32" isArray="false" followOnExport="false"
  isNullsAllowed="true" isUnique="false" isPublicRead="false"
  isPublicWrite="false" isCandidateKey="false" arrayLength="0"
  isTransient="false" exportAsString="false" noBackpointer="false"
  initialValue="NULL"/> </TcAttributeAttach>
  <TcGRMRule primaryTypeName="CAEAnalysisRevision "
  secondaryTypeName="CAEResultRevision" relationTypeName="TC_CAE_Results"
  primaryCardinality="1" secondaryCardinality="1" secured="false"
  attachability="WriteAccessReq" changeability="Changeable"
  detachability="WriteAccessReq"/>
</Delete>
```

There are two elements in the **Delete** element. The first is the **CAEAnalysisRevMasterStore** element. This element represents a class with a **solution_step** attribute that is missing on the class. Either add the attribute in the installation/upgrade script or remove the definitions from the XML template. The second element is a GRM rule. The GRM rule has been removed from the database. Therefore, either add a command to add the GRM rule, or remove the XML definition of the GRM rule from the template.

Resolving Change element issues

Any element that appears in the **Change** element exists in both the new model and in the old model. However, one of the features of the definition has been changed. In this case, the **Change** elements are in the extracted database model in one state and the expected template model has it in another state. Therefore, you must either add a command in the installation/upgrade script to reconcile the difference, or the XML template definitions must be corrected.

For example, a change file has the following **Change** element:

```
<Change><TcAttributeAttach className="Architecture">
  <TcAttribute attributeName="has_basedon_preexist_elemnt"
  attributeType="POM_logical" maxStringLength="0" isArray="false"
```

```

followOnExport="false" isNullsAllowed="false" isUnique="false"
isPublicRead="false" isPublicWrite="false" isCandidateKey="false"
arrayLength="0" isTransient="false" exportAsString="false"
noBackpointer="false"/>
</TcAttributeAttach>

</Change>

```

When a **<Change>** element is written, the entire element is written to the file not just the changed feature of the element. Therefore, to understand what has changed, you must compare the changed element to the same element definition in the expected XML template file. The following shows the same definition in the XML template file:

```

<TcAttribute arrayLength="0" attributeName="has_basedon_preexist_elemnt"
  attributeType="POM_logical" exportAsString="false"
followOnExport="false"
  initialValue="false" isArray="false" isCandidateKey="false"
  isNullsAllowed="true" isPublicRead="true" isPublicWrite="false"
  isTransient="false" isUnique="false" maxStringLength="0"
  noBackpointer="false"/>

```

The **isNullsAllowed** attribute for the **has_basedone_preexist_elemnt** attribute has been changed. You must either add a command to reconcile the difference or the XML template must be corrected to match the database.

Change elements that include TcTool elements

Ignore any **Change** elements that include **TcTool** elements. Tools are stored in the XML with no release date. However, when they are installed into the database, they are automatically assigned a date. The comparison reveals that there is a date change between the original definition and the extracted definition.

Update the site templates and then apply these templates. Otherwise, changes made to the site are lost in subsequent upgrades.

The example in [Update the target site schema](#) shows how you can use the **database_verify** utility and Business Modeler IDE capabilities to define the necessary schema changes. Because this was done in a lab environment, the deltas were applied directly and templates were not modified. Siemens Digital Industries Software does not recommend this for a production environment because, if you do not maintain the templates, the changes can be lost in subsequent upgrades.

Update the target site schema

For site consolidation, you move and consolidate data between a source site and a target site. The two database schemas must be similar before a successful transfer of bulk data is possible. The process consists of several steps using different tools. The first step is to find out the differences by running the **database_verify** utility. This utility compares the two database schemas, looking for Teamcenter types,

tools, release statuses, and units of measure objects between two specified Multi-Site Collaboration sites and generates a report of any database discrepancies. The second step uses the **bmide_comparator** utility.

You use the Business Modeler IDE to customize Teamcenter. Through the Business Modeler IDE interface you can add types, properties, and methods modules. Use the **bmide_comparator** utility to compare two complete Teamcenter model files and generate a differences (-delta) XML file. You can then use this differences file to upload or create what is missing in the target database. For a production operation, it is important to update the target site Business Modeler IDE templates so that future upgrades do not lose the changes. Both tools must be run in a Teamcenter environment shell.

1. Use **database_verify** utility to determine differences:

```
database_verify -u=tc-admin-user -p=password -g=group
               -v -from=target -to=source -all -output=outputfilename.txt
```

Place the output file information in a spreadsheet that you can use to compare the differences.

2. Use the **bmide_comparator** utility to get the database discrepancies:

```
bmide_comparator -all -old= model_site1.xml -new
model_site2.xml -delta =site1tosite2_delta.xml
-log=bmide_comparator.log
```

The output file contains the differences. Siemens Digital Industries Software recommends that you remove all elements except the **<Add>** elements from the file. This lets you add missing objects to the target site schema so that it contains the required source site objects.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Siemens and the Siemens logo are registered trademarks of Siemens AG.
UGS, Teamcenter and UGS Teamcenter are trademarks or registered trademarks of UGS or its subsidiaries.
This software and related documentation are proprietary to UGS Corp. (c)2007 UGS Corp. All rights reserved.
-->
<TcBusinessData xmlns="http://teamcenter.com/BusinessModel/TcBusinessData"
Date="Tue Sep 15 03:28:41 PM 2009" TcVersion="">
  <Add>
    <TcClass className="PartIdentity" parentClassName="POM_Object" isExportable="false"
isUninstantiable="false" isUninheritable="false">
      <TcAttribute attributeName="partNumberKey" attributeType="POM_string" maxStringLength="128"
isArray="true" followOnExport="false" isNullsAllowed="true" isUnique="false"
isPublicRead="false" isPublicWrite="false" isCandidateKey="false" arrayLength="-1"
isTransient="false" exportAsString="false" noBackpointer="false"/>
      <TcAttribute attributeName="partNumberValue" attributeType="POM_string" maxStringLength="128"
isArray="true" followOnExport="false" isNullsAllowed="true" isUnique="false"
isPublicRead="false" isPublicWrite="false" isCandidateKey="false" arrayLength="-1"
"justadds.xml" 2115 lines, 155991 characters
```

```
<?xml version="1.0" encoding="UTF-8"?>
:
<tcBusinessData xmlns="http://teamcenter.com/BusinessModel/TcBusinessData"
Date="Tue Sep 15 03:28:41 PM 2009" TcVersion="">
  <Add>
```

3. Upload the edited differences (**<Add>** elements only) file using the **business_model_updater** utility.

```
business_model_updater -u=tc-admin-user -p=password -g=group
  -mode=upgrade -update=all -process=add
  -file=justadds.xml -log=justadds.log
```

```
-----
Update Option : all
Log File Name : ./justadds2.log
Model File Name : ./justadds.xml
Run Mode : upgrade
Process Option : add
-----
```

```
-----|-----|-----|-----|-----
Operation| Element Type | Element Name | ErrorNo| Error
-----|-----|-----|-----|-----
```

```
Processed Schema Changes:
```

```
-----|-----|-----|-----|-----
Add | TcClass | PartIdentity | 0 | Success
...
Add | TcClass | infoClass | 0 | Success
Add | TcClass | pm_master | 0 | Success
Add | TcClass | pm_rev_master | 0 | Success
Add | TcClass | productState | 0 | Success
Add | TcClass | purposeClass | 0 | Success
Add | TcClass | regNumClass | 0 | Success
Add | TcClass | req_specs_master | 0 | Success
Add | TcClass | req_specs_rev_master | 0 | Success
```

```
...
Add | TcClass | statusClass | 0 | Success
Add | TcClass | typeInfoClass | 0 | Success
Add | TcClass | validation_plan_master | 0 | Success
Add | TcClass | validation_plan_rev_master | 0 | Success
Add | TcClass | constraintClass | 0 | Success
```

```
-----|-----|-----|-----|-----
Unprocessed Schema Changes:
```

```
-----|-----|-----|-----|-----
Processed Non-Schema Changes:
```

```
-----|-----|-----|-----|-----
Add | TcStandardType | COSTVariantRelation | 0 | Success
Add | TcStandardType | DTYPE_new_state | 0 | Success
Add | TcStandardType | DTYPE_obselete_state | 0 | Success
Add | TcStandardType | DTYPE_old_state | 0 | Success
Add | TcStandardType | DTYPE_reg_num_rel | 0 | Success
...
Add | TcStandardType | validation_plan_rev_master | 0 | Success
Add | TcForm | Build Request Form | 0 | Success
Add | TcForm | Config Component Part | 0 | Success
Add | TcForm | Material Request Form | 0 | Success
...
Add | TcForm | validation_plan Master | 0 | Success
Add | TcForm | validation_plan Revision Master | 0 | Success
```

```
...
Add | TcStatus | 404_1 | 0 | Success
Add | TcStatus | 404_1B | 0 | Success
Add | TcStatus | 404_1PF | 0 | Success
Add | TcStatus | 404_1PR | 0 | Success
Add | TcStatus | 404_2 | 0 | Success
Add | TcStatus | 404_2B | 0 | Success
```

```
...
Add | TcStatus | X_5PF | 0 | Success
Add | TcStatus | X_5PR | 0 | Success
```

```

Add |TcTool |CGM_Dummy_Tool | 0|Success
Add |TcTool |HPGL_Dummy_Tool | 0|Success
Add |TcTool |TIF_Dummy_Tool | 0|Success
Add |TcTool |cgm_view | 0|Success
Add |TcTool |des2table | 0|Success
Add |TcTool |ug_to_vda | 0|Success
Add |TcTool |vda_to_ug | 0|Success
Add |TcToolInputForma|TIFView:BINARY | 0|Success
|tAttach | | |
Add |TcToolOutputForm|TIFView:BINARY | 0|Success
|atAttach | | |
Add |TcDataset |Excel | 0|Success
Add |TcDataset |ExcelTemplate | 0|Success
Add |TcDataset |Powerpoint | 0|Success
Add |TcDataset |PowerpointTemplate | 0|Success
Add |TcDataset |Word | 0|Success
Add |TcDataset |WordTemplate | 0|Success
Add |TcDatasetToolAtt|DirectModel | 0|Success
|ach | | |
...
Add |TcNoteType |@ARCSET | 0|Success
Add |TcNoteType |@EngCD001_001 | 0|Success
Add |TcNoteType |@INTSET | 0|Success
...
Add |TcStorageMedia |archive_tdc1 | 0|Success
Add |TcTypeDisplayRul|CM Context:Organization | 0|Success
|e | | |
Add |TcTypeDisplayRul|CM Context Job:Organization | 0|Success
|e | | |
...
Add |TcGRMRule |CORP_Part Revision:Image:IMAN_ma| 0|Success
| |nifestation | |
Add |TcGRMRule |CORP_Simulation Revision:DirectM| -1|CORP_Simulation
Revision ( DirectModelMotion ) a | |odelMotion:IMAN_manifestation | |lready exists
Add |TcGRMRule |CORP_Software Revision:Dataset | -1|CORP_Software
Revision ( Dataset ) already exist | | | |s
...
Add |TcLOV |Common Engineering Form | 0|Success
Add |TcLOV |Dwg Emission Depths | 0|Success
Add |TcLOV |PAD Element Instruction1 | 0|Success
Add |TcLOVValueAttach|Dataset Context Common1 | 0|Success
...
| |Master | |
-----|-----|-----|-----|-----
Unprocessed Non-Schema Changes:
-----|-----|-----|-----|-----
Add |TcLOVValueAttach|OverAllScore_OverAllDirector | 54024|LOV has corrupted data
Add |TcLOVValueAttach|OverAllScore_OverAllDirector | 54024|LOV has corrupted data
Add |TcLOVValueAttach|OverAllScore_OverAllDirector | 54024|LOV has corrupted data
...
|Attach |Static | |he Change Type "GM Product Change Order"
Add |TcChangeIdFormat|PWT Common Change:1-3:APG:Static| -1|ID Format "1-3:APG:Static"
already attached to t
|Attach | | |he Change
...
Add |TcNamingRuleAtta|CORP_Equipment:item_id: ItemRu| 74004|This property already has
a Name Rule attached
|ch |le | |
Add |TcNamingRuleAtta|CORP_Install:item_id: ItemRule| 74004|This property already has
a Name Rule attached

```

```

...
Add |TcNamingRuleAtta|CORP_Simulation:item_id: ItemR| 74004|This property already has
  a Name Rule attached
  |ch |ule | |
Add |TcNamingRuleAtta|CORP_Vehicle:item_id: ItemRule| 74004|This property already has
  a Name Rule attached
  |ch | | |
Memory=15060574, SQL=14692, Time 16.790000s cpu, 212.138347s real at-Statistics

```

After you add the differences to the target database, you have similar schemas for a more reliable import.

Planning increments

Data partitions

Use increment planning to partition and map the data to be moved.

A *partition* is a logical segmentation of the source site product data from a business point of view, such as the data for a specific product program, the released data for a vehicle program, or a library of standard parts. Normally, one or more complete partitions of data is moved during a single execution increment. In the case of a divestiture, it may be all the data to be retained by the divesting enterprise or all the items to be given to the divested organization. The one criteria that all partitions must meet is that they must be able to be defined to the site consolidation tools.

Creating partitions

The site consolidation tools provide the following flexibility in defining the objects to be consolidated:

- **ItemId** can specify a single specific item ID or a template to specify items in the partition. This approach can be useful if naming conventions are well defined and strictly followed.
- **Class** can be used in conjunction with name or input file to identify that the list of names provided only apply to a specific class of workspace object.
- **Name** can be used in conjunction with **Class** to specify a single workspace object.
- An input file can be used to provide a list of names. If these are not the names of items, **Class** must be used.

Typically, it is best to move the data that is not dependent on other data first. This minimizes activities such as stub creation to satisfy references to data to be moved later.

1. Determine the major product programs that are to be moved in the various execution increment. Each of these programs represents a partition.

2. Determine the data, such as standard parts libraries, the product programs in the first increment are dependent on. For example, groups of standard electronic components, groups of standard mechanical connectors, and so forth. Each of these groupings represents another partition.
3. Move the standard parts partition first and then move the product program. This minimizes stubbing and then replacing stubs, which can have a significant impact on performance.

Developing a detailed transfer plan

Typically, Teamcenter structures are large and complex. There are scalability limits imposed on the site consolidate (SC) tools by environment considerations. For example, on Windows systems, process sizes cannot exceed 3 GB. There can also be time constraints imposed on the execution of site consolidation tools. For example, they can only be run in off-hours and are allowed to run for no more than 12 hours on any given day. To address this issue, Siemens Digital Industries Software recommends that data being consolidated be investigated in a lab environment. Siemens Digital Industries Software also recommends that the lab environment structurally mirror the production environment with respect to the following:

- Source and target databases are instantiated in the lab, including volumes. The actual volume files can be created as zero length files.
- At least one third site should also be instantiated in the lab environment.
- If the copied databases are in a Multi-Site federation in production, this should also be mirrored in the lab environment.
- The tools are input/output (I/O) intensive so the lab should have good I/O subsystems.

After you create the lab environment, the data can be investigated using the **plm_report_extract** utility, some of the underlying orchestration components, lightweight closure rules, and default closure rules.

1. Evaluate your environment to understand sharing patterns and the quantities of data. This can be done in the production environment. However, it is less complex to look at the more static environment of the lab.
2. In general, site consolidation closure rules and their controlling transfer option sets are used to provide direction to the site consolidation tools on how to traverse the Teamcenter database and whether to process the information traversed. Teamcenter provides a set of lightweight closure rules and a set of default closure rules as part of the install and upgrade process.

Caution:

Before performing site consolidation execution phase activities, you must remove any references to obsolete attributes and classes from custom closure rules or property sets. Additionally, you should remove any deprecated attributes and classes from custom closure rules or property sets.

For information about deprecated and obsolete classes and attributes, see the support announcements in the latest Teamcenter **README** file.

Following are standard site consolidation transfer option sets and closure rules.

SiteConsolidationDefault transfer option set

Option	Display name	Default value	Description	Group name
opt_res_checkout	Include ImanReservation	False	Export ImanReservation for checked-out workspace object	Relations
opt_res_audit	Include ImanFile	True	Export ImanFile for checked out Workspace object	Relations
opt_entire_bom	Export entire BOM	True	Export all components if the item selected is an assembly	Product Structure Options
opt_entire_mse	Export BOM BOP work area structure together	False	Export BOM BOP work area structure together	Relations
internalClosureRule	User internal closure rules	True	Use internal closure rules for traversal to determine required helper objects	General Options
opt_mechatro	Mechatronics	False	Export Mechatronics data	Mechatronics

SiteConsolidationDefaultCR closure rule

Primary object class	Primary object	Secondary object class	Secondary object	Relation type	Related property or object	Action type	Conditional clause
Class	MEProcessRevision	Class	ItemRevision	RELATIONSP2S	IMAN_METarget	SKIP	\$opt_entire_mse="false"
Class	MEProcessRevision	Class	MEProcessRevision	RELATIONSP2S	IMAN_MEWorkArea	SKIP	\$opt_entire_mse="false"
Class	ItemRevision	Class	ItemRevision	RELATIONSP2S	*	PROCESS+TRAVERSE	
Class	ItemRevision	Class	Item	RELATIONSP2S	*	PROCESS+TRAVERSE	
Class	WorkspaceObject	Class	WorkspaceObject	RELATIONSP2S	*	PROCESS+TRAVERSE	

SiteConsolidationLW transfer option set

Option	Display name	Default value	Description	Group name
opt_res_checkout	Include ImanReservation	False	Export ImanReservation for checked out Workspace object	Relations
opt_res_audit	Include ImanFile	True	Export ImanFile for checked out Workspace object	Relations
opt_entire_bom	Export entire BOM	True	Export all components if the item selected is an assembly	Product Structure Options
opt_entire_mse	Export BOM BOP WorkArea Structure together	False	Export BOM BOP WorkArea Structure together	Relations
internalClosureRule	User Internal Closure Rules	False	Use internal closure rules for traversal to determine required helper objects	General Options

SiteConsolidationLWCR closure rule

Primary object class	Primary object	Secondary object class	Secondary object	Relation type	Related property or object	Action type
Class	Item	Class	ItemRevision	REFBY	items_tag	TRAVERSE
Class	ItemRevision	Class	PSBOMViewRevision	ATTRIBUTE	structure_revision	TRAVERSE
Class	PSBOMViewRevision	Class	PSOccurrence	REFBY	parent_bvr	TRAVERSE
Class	PSOccurrence	Class	WorkspaceObject	ATTRIBUTE	child_item	PROCESS+ TRAVERSE

Improving plm_report_extract utility performance

Due to scalability and performance issues, you may need to use the **plm_report_extract** utility with the **SiteConsolidationLWCR** option set to list out the approximate number of components in the given assembly and then use the **SiteConsolidationDefaultCR** option set with the **\$opt_entire_mse** condition set to **false** (this is the default value) to traverse the list of components to determine the objects under the given top level object.

The performance of the utility with these options and separate runs with each closure rule is much faster when compared to running traversal for the top level object with the **SiteConsolidationDefaultCR** option set with the entire BOM traversal (**\$opt_entire_mse** condition) set to **true**. You must be aware that you do not get the same list of objects using the separate runs as compared to using the **SiteConsolidationDefaultCR** option set with the entire BOM traversal (**\$opt_entire_mse** condition) set to **true**. Some of the objects are missing, however the missing objects can be transferred as needed,

or transferred during the final run of the `sitcons_accountability_chk` utility, when all objects that are missing during replication are transferred.

The `cpdSiteConsInternalClosureRules` closure rule provides support for traversing 4th Generation Design (4GD) objects for site consolidation actions.

Run the `plm_report_extract` utility with lightweight closure rules

Caution:

Before you run the `plm_report_extract` utility with the lightweight closure rules (`SiteConsolidationLW`), ensure the `internalcluserule` option is set to **false**. This is the default value for this option, however if it is set to **true** it can cause the utility to go into an infinite loop.

1. Run the `plm_report_extract` utility and the lightweight closure rules with their transfer options as in the following example. (The `plm_report_extract` utility can also be used in the production environment because it does not modify the databases.)

```
plm_report_extract -u=tc-admin-user -p=password -g=group -verbose
-inputfile=D:\JCF_PN\cp_g1\dir_6TO15\cp_g1_i6TO15.txt
-output_file=D:\JCF_PN\cp_g1\dir_6TO15\cp_g1_i6TO15_lwcr_ext.bin
-attributes_file=F:\PN\tcdatapn\default_attributes_file.txt
-optionset=SiteConsolidationLWCR

Tue 04/15/2021
07:21 PM
Parsing the attributes file
Traversing the dependent objects
=====
=      Total No. of primary objects : [10]      =
=      Total No. of object traversed : [264]    =
=====
Time taken for traversal: 0.671000s cpu, 13.156000s real
Serializing the header information
Serializing basic information
Serializing the objects
Time taken for query and serialize: 0.031000s cpu, 10.078000s real
"Stopping LW Extract"
Tue 04/15/2021
07:22 PM
```

From this output, extract items of the type you want to transfer to create the set of objects in the BOM of the initial input items. You can then use the extracted items file as input to the traversal using the default closure rules with BOM traversal off.

2. Run of the `plm_report_extract` utility with entire BOM traversal turned off as in the following example:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -verbose
-inputfile=D:\JCF_PN\cp_g1\dir_6to15\cp_g1_i6to15_dfcr.txt
-output_file=D:\JCF_PN\cp_g1\dir_6to15\ext_plmr_dfcr\cp_g1_i6to15_
dfcr_ext.bin
```

```
-attributes_file=TC_DATA\default_attributes_file.txt

Tue 04/15/2021
07:46 PM
Parsing the attributes file
Traversing the dependent objects
=====
=      Total No. of primary objects : [125]      =
=      Total No. of object traversed : [43191]   =
=====
Time taken for traversal: 5.858000s cpu, 232.884000s real
Serializing the header information
Serializing basic information
Serializing the objects
Time taken for query and serialize: 4.953000s cpu, 238.134000s real
"Stopping Default Extract"
Tue 04/15/2021
07:54 PM
```

Create a report with units of measure

If you require unit of measure (UOM) data, such as Unit or Symbol in the report, perform the following steps before running the **plm_report_extract** utility:

1. Add the following to the **default_attr_file.txt** file:

```
Symbol:UnitOfMeasure,symbol
Unit:UnitOfMeasure,unit
```

2. In the PLM XML/TC XML Export Import Administration application, navigate to the **SiteConsolidationDefault** transfer option set and add the **opt_plm_report** option with the **Default Value** set to **True** and save the option set.
3. Navigate to the **FT_Item** closure rule and add the following clause and save it:

```
CLASS.Item:CLASS.UnitOfMeasure:ATTRIBUTE.uom_tag:PROCESS:$opt_plm_report=="true"
```

Run the **plm_report_extract** utility and the output report includes the UOM attributes.

Caution:

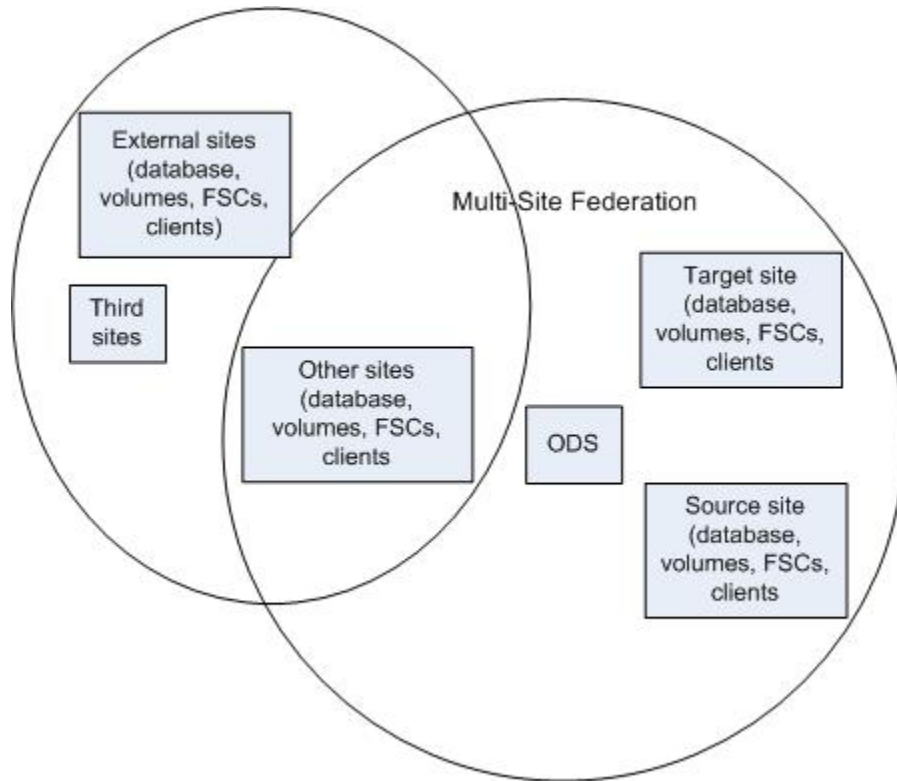
This condition must be set to **False** before you run the **tcxml_export** utility.

Detecting and repairing data anomalies

Detection and repair tools

In the preparation phase, you must conduct in-depth product data analysis using the PLM report utilities provided with site consolidation. These tools are designed to help you manage conflict resolution,

ownership issues, requirements from outcome of high level plan and site audits, and data repair in a Multi-Site federation.



Replace invalid characters

Invalid characters in a database are characters that are not part of the supported character set of the database. For example, attempting to put multibyte characters into an ASCII database can result in invalid characters. You should identify and clean up such character strings in the preparation phase in the source database.

1. Run **plm_report_extract** and generate the uncompressed report.
2. Use a text editor that can handle large file sizes to search for invalid characters.
3. Identify the affected object.

For example, the description of an item may contain an invalid character.

4. Use the rich client to replace the invalid characters with a suitable equivalent.

Detect and correct any source site references to CFM configuration object data

If your source database has **PCFM_configuration_object** references, you must correct these prior to consolidating the sites.

1. Use the following SQL query at the source site to determine if **PCFM_configuration_object** references exist in the database:

```
select count(*) from pitem
where RCONFIGURATION_OBJECT_TAGU != 'AAAAAAAAAAAAAAAA';
```

2. If this command returns more than zero objects, you must clean up the data using the following SQL commands:

Caution:

These SQL commands change the Teamcenter database. Siemens Digital Industries Software recommends that you perform the commands on a test copy of the production data initially to confirm that it addresses the issue. Siemens Digital Industries Software also recommends that you create a backup/snapshot of the production database before using these commands on that database.

```
select count(*) from pom_backpointer,
pitem where RCONFIGURATION_OBJECT_TAGU != 'AAAAAAAAAAAAAAAA' and
from_uid = pitem.puid and to_uid = rconfiguration_object_tagu;

select count(*) from pcfm_configuration_object;

delete from pom_backpointer where exists (select puid from
pitem where RCONFIGURATION_OBJECT_TAGU != 'AAAAAAAAAAAAAAAA'
and from_uid = pitem.puid and to_uid = rconfiguration_object_tagu);

update pitem set rconfiguration_object_tagu='AAAAAAAAAAAAAAAA'
where rconfiguration_object_tagu != 'AAAAAAAAAAAAAAAA';

commit;
```

Missing file data at the source site

While not a critical issue for the site consolidation tools, the best time to address this issue is before the creation of the shared volume environment.

The **review_volumes** tool can be run to generate reports on the source site volume storage and can provide additional information to help identify and locate the files.

When the source and target site are operating in a shared volume mode, **review_volumes** should not be used for more than volume reporting at either the source or target site. These files must either be restored from the customer backups or, if they were removed by that tool and the archive option was used, from **review_volumes** archives.

Managing intersite conditions

Intersite data preconditions

To manage intersite conditions, you need to know the production status of your data, whether it is in your federation, whether it has anomalies, what volumes house the data, whether FCSs are deployed, and whether C/S or S/S network connectivity is present.

Intersite data precondition matrix			
Site	In production	In Multi-Site federation	Has relevant data anomalies
Source	Yes	Yes	Yes
Target	Yes	Yes	Yes
Other	Yes	Yes	No
ODS	Yes	Yes	Yes
External	Yes	No	No

Capabilities of the PLM report utilities

The PLM report utilities capabilities are specifically targeted at site consolidation although they can be useful in other contexts. For site consolidation, the utilities provide:

- The capability to extract the necessary information from the relevant sites. Emphasis is placed on the site containing the information to be consolidated (source) and the site to which the information is to be consolidated (target).
- The capability to holistically analyze the extracts for issues such as ownership inconsistencies, uniqueness conflicts across sites, and other data error conditions relevant to site consolidation.
- Guidance on how to resolve the issues identified.
- A preview capability to examine data to be moved.

There are three utilities:

- **plm_report_extract**

Extracts persistent data from the Teamcenter database and stores it in intermediate binary (not human readable) format. The extract is run independently for each site.

The **plm_report_extract** utility takes a *bottom up* approach to looking at items and objects for information gathering and analysis to provide the perspective you need to structure consolidation.

- **plm_report_consistency_analysis**

Using the site extracts, this utility analyzes, reports, and suggests fixes for internal and cross site consistency issues relevant to site consolidation. Unlike the **plm_report_extract** utility, this utility produces reports in user readable format.

Be aware that this utility uses the pipe character (vertical bar - |) as a delimiter for creating the uncompressed report. If an object's descriptive data, such as the **Description** attribute, contains a pipe operator, the uncompressed output is distorted. This can also happen if newline characters are present in the descriptive data.

This is an example of the report output in a text editor:

```
Consistency Analysis Report
Inconsistency Analysis Report Created at [2013-03-06 15:21:49]
Site ID=[137512324] Site Name=[bklair] - Extraction Date: [2013-03-06 14:44:09] Extract File
path=[/sc1/
tc10.1_scripts/unix_target/dir_plm_rpt/12578339/dir_extracts/12578339_bkla_dfc_r_ext.bin]
Site ID=[491361009] Site Name=[clteng] - Extraction Date: [2013-03-06 14:44:15] Extract File
path=[/sc1/
tc10.1_scripts/unix_target/dir_plm_rpt/12578339/dir_extracts/12578339_cin2_dfc_r_ext.bin]

UID|Name|Level|Type/Class|Parent|Bucket|Site Ownership|Object Published to ODS|Publishing
Site|Object Pre
sent at Sites|Sites Object Exported To|Checked Out Users|Checked Out Sites|Has SST|Potential
Issue|Severi
ty|Suggestion Fixup
AR80dHSTgDNxsC|12578339.geo_fin001.001999.prt|1|Item Revision Alias||-1|cin2ct,cin2ct|No||
bklair,clteng||
||No|Missing Export Records, |3||
AVz0dHSTgDNxsC|t3ddd12e8.tmp|2|ImanFile|AR80dHSTgDNxsC|-1|cin2ct,cin2ct|No||bklair,clteng|||
No|Missing E
xport Records, |3||
Al_wcVPNgDNxsC|PF117|1|GM Work Order||-1|0,cin2ct|No||clteng|||No|Object missing w.r.t
parent, Missing E
xport Records, |1||
BY3wvVGngDNxsC|t3cf66a65.tmp|2|ImanFile|Al_wcVPNgDNxsC|-1|0,cin2ct|No||clteng|||No|Object
missing w.r.t
parent, Missing Export Records, |1||
QGywcVPNgDNxsC|Affected Items|2|ImanRelation|Al_wcVPNgDNxsC|-1|0,cin2ct|No||clteng|||No|
Object missing w
.r.t parent, Missing Export Records, |1||
QKzwcVPNgDNxsC|Affected Items|2|ImanRelation|Al_wcVPNgDNxsC|-1|0,cin2ct|No||clteng|||No|
Object missing w
.r.t parent, Missing Export Records, |1||
g17wcVPNgDNxsC||2|ReleaseStatus|Al_wcVPNgDNxsC|-1|0,cin2ct|No||clteng|||No|Object missing
w.r.t parent,
```





```

Missing Export Records, |1||
gB$wcVPNgDNxsC|Affected Items|2|ImanRelation|Al_wcVPNgDNxsC|-1|0,cin2ct|No||clteng|||No|
Object missing w
.r.t parent, Missing Export Records, |1||
Au0w7$tKgDNxsC|SLEEVE-CYL (MCHG)|1|CORP_Part||-1|cin2ct,cin2ct|No||bklair,clteng|||No|
Missing Export Rec
ords, |3||
3ALdd4LpAfcCdC||2|ProjectObjectRelation|Au0w7$tKgDNxsC|-1|0,clteng|No||clteng|||No|Object
missing w.r.t
parent, |1||
AH804DWngDNxsC|SLEEVE-CYL (MCHG)/003|2|CORP_Part Revision|Au0w7$tKgDNxsC|-1|0,cin2ct|No||
clteng|||No|Obj
ect missing w.r.t parent, Missing Export Records, |1||

```

Importing the output into an Excel spreadsheet improves readability:

3. Preparing for site consolidation

	A	B	C	D	E	F	G	H	
1	Consistency Analysis Report								
2	Inconsistency Analysis Report Created at [2013-03-06 15:21:49]								
3	Site ID=[137512324] Site Name=[bklair] - Extraction Date: [2013-03-06 14:44:09] Extract File path=[/sc1/tc10.1_scripts/uni								
4	Site ID=[491361009] Site Name=[clteng] - Extraction Date: [2013-03-06 14:44:15] Extract File path=[/sc1/tc10.1_scripts/uni								
5									
6	UID	Name	Level	Type/Class	Parent	Bucket	Site Ownership	Published to ODS	F
7	AR80dHSTgDNxsC	12395783.geo_fin001.001999.prt	1	Item Revision Alias		-1	cin2ct,cin2ct	No	
8	AVz0dHSTgDNxsC	t3ddd12e8.tmp	2	ImanFile	AR80dHSTgDNxsC	-1	cin2ct,cin2ct	No	
9	Al_wcVPNgDNxsC	PF117	1	GM Work Order		-1	0,cin2ct	No	
10	BY3wvVGngDNxsC	t3cf66a65.tmp	2	ImanFile	Al_wcVPNgDNxsC	-1	0,cin2ct	No	
11	QGywcVPNgDNxsC	Affected Items	2	ImanRelation	Al_wcVPNgDNxsC	-1	0,cin2ct	No	
12	QKzwcVPNgDNxsC	Affected Items	2	ImanRelation	Al_wcVPNgDNxsC	-1	0,cin2ct	No	
13	g17wcVPNgDNxsC		2	ReleaseStatus	Al_wcVPNgDNxsC	-1	0,cin2ct	No	
14	gB\$wcVPNgDNxsC	Affected Items	2	ImanRelation	Al_wcVPNgDNxsC	-1	0,cin2ct	No	
15	Au0w7\$tKgDNxsC	SLEEVE-CYL (MCHG)	1	CORP_Part		-1	cin2ct,cin2ct	No	
16	3ALdd4LpAfcCdC		2	ProjectObjectRelatio	Au0w7\$tKgDNxsC	-1	0,clteng	No	
17	AH804DWngDNxsC	SLEEVE-CYL (MCHG)/003	2	CORP_Part Revision	Au0w7\$tKgDNxsC	-1	0,cin2ct	No	
18	A3504DWngDNxsC		3	ReleaseStatus	AH804DWngDNxsC	-1	0,cin2ct	No	
19	AL_04DWngDNxsC	IMAN_master_form	3	ImanRelation	AH804DWngDNxsC	-1	0,cin2ct	No	
20	AL204DWngDNxsC	12395783/003	4	CORP_Part Revision Master	AL_04DWngDNxsC	-1	0,cin2ct	No	
21	AL504DWngDNxsC		5	CORP_PartRevMaster	AL204DWngDNxsC	-1	0,cin2ct	No	
22	gD304DWngDNxsC	IMAN_RES_audit	5	ImanRelation	AL204DWngDNxsC	-1	0,cin2ct	No	
23	gD004DWngDNxsC	cico_audit_file_3d8f2205.txt	6	ImanFile	gD304DWngDNxsC	-1	0,cin2ct	No	
24	ALx04DWngDNxsC	IMAN_based_on	3	ImanRelation	AH804DWngDNxsC	-1	0,cin2ct	No	
	Ahv04DWngDNxsC	IMAN specification	3	ImanRelation	AH804DWngDNxsC	-1	0,cin2ct	No	

Inconsistency Analysis Report 2

Ready

An object in an island-of-data was reported in the extracts as not being owned by a site. This may not be an issue since the owning site's extract may not have been an available input.

- **Multiple Master Sites — Multiple Site Ownership**

An object in an island-of-data was reported in the extracts as being owned by more than one site.

- **Ownership Error — Ownership inconsistent with parent**

An object in an island-of-data has an owning site different than its parent object in the same island-of-data. An example of this is the Item and item revision at a site having different owning sites.

- **Object missing with respect to (w.r.t) parent**

Within an island-of-data, an object required by the database schema is missing from a site.

Severity 2: The error message contains one of the following values:

- **Object Publication Inconsistent With ODS — Obsolete Publication**

The site publishing the object to the ODS does not own the published object.

- **Export Record At Multiple Sites**

Export records exists at more than one site for an object in the island of data.

- **Publication Record Information Not available —**

A cross-site condition in which an object in an island-of-data exists as at another site, but there is no export record.

Severity 3: The error message contains one of the following values:

- **Missing Export Records**

A cross-site condition in which an object in an island-of-data exists as at another site, but there is no export record.

- **Export Site Information Not available**

- **Obsolete Export Records**

An export record for an object in the island-of-data indicates that an object was exported to another site, but the object does not exist at the other site.

Additional messages that may be reported in the **Potential Issues** column include:

- **Objects with SST datasets**

One or more objects in the island-of-data have an associated synchronous site transfer (SST) recovery dataset. Use the **ensure_site_consistency** utility on the object to take corrective action.

- **Checked out status**

An object in an island-of-data is checked out. It is not an error condition, but ownership transfer requires that no object in the island-of-data be checked out.

- **Inconsistent island-of-data definition across multiple sites**

A cross site condition in which an island-of-data contains different objects in the extracts from the different sites.

The **Level** and **Bucket** columns are for internal use only by Siemens Digital Industries Software development.

- **plm_report_constraint_analysis**

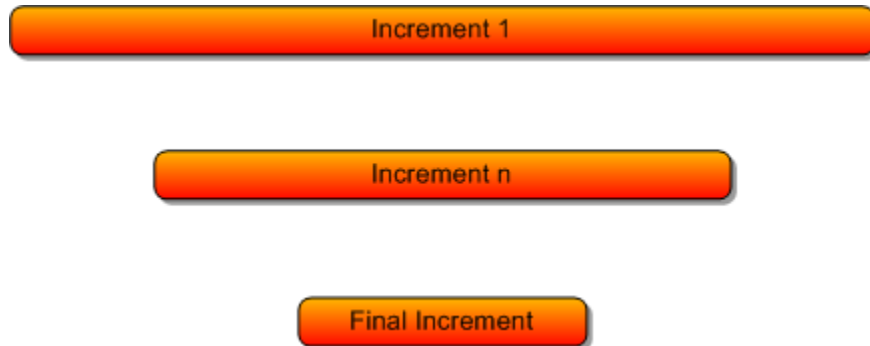
Using the source site extracts, this report analyzes the target site and reports database-level constraint violations that can arise when the import tools attempts to import the data to the target site provided by the export tools. The source site's extract file generated by the **plm_report_extract** utility drives this analysis.

4. Execution phase activities

Execution phase increments

You move product data from the source site to the target site during the execution phase.

The execution phase is performed in one or more increments.



During each increment, one or more partitions of data is replicated and then ownership is reassigned from the source to the target site.

References in the source data being imported are mapped to the shared volumes by the replication function.

Volume data is not moved during this phase.

Replicating data and changing ownership in a typical execution phase increment

The typical execution phase increment consists of a replication period followed by the *time critical period* (TCP) in which ownership change is performed for the product data that was replicated.

- Replication phase steps:
 1. Replicate product metadata.
 2. Synchronize data (`sitcons_replicate_mgr -sync`).
 3. Evaluate replication.
- Time critical period steps:
 1. Disable user access.

2. Back up data.
3. Perform final synchronization (`sitcons_replicate_mgr -sync`).
4. Validation replication.
5. Change ownership of data.
6. Evaluate the target site.
7. Evaluate the source site.

File system cache (FSC) prepopulation is not impacted by site consolidation. Cached objects are identified to the FSC by GUIDs and are not attached to the original database.

Consolidating using multiple execution phase increments

You may want to consolidate your sites using more than one execution phase increment. For each increment you have a replication period followed by a time critical period (TCP). The following sequence is the recommended process for using multiple increments:

1. Replication phase 1.
2. TCP 1.
3. Replication phase 2.
4. TCP 2.
5. Accountability check (using `sitcons_accountability_chk` utility).
6. Additional replication phases and TCPs as required by accountability check results.
7. Clean up phase.

This process ensures that any non-consolidated objects reported by the `sitcons_accountability_chk` utility, which are required at the target site, do not get cleaned up with the source site.

Replication phase

Replication phase activities

The *replication phase* is the first phase of any execution increment. In this phase, you use *fast replication* for source site export, mapping, and target site import of the product data as replicas. These actions occur in multiple steps.

The capability to perform replication is required to support the large databases that exist at many client sites.

Fast replication:

- Reflects the conventions of classic Multi-Site designed to ensure data integrity.
- Does not invalidate the concurrent use of existing Multi-Site data sharing operations or of other production operations at the source and target sites.

Fast replication differs from standard Multi-Site operations in the following ways:

- Replicas at the source site are replicated from the source site to the target site.
- Export records are replicated to the target site.
- Export records are not created at the source site for owned data moved as replicas. Subsequent updates of objects replicated by the site consolidation tools are based on the accountability table and the object's modification data.
- Evaluation is conducted from the administration perspective.
- After consolidation data is initially replicated. Export records are not used to identify source site updates to previously replicated data to the target site.

Replication and fast synchronization (fast sync) operations are iterative and must be repeated until all the data in the increment is updated. The following table represents a prototypical site consolidation replication phase, including the primary tools used, the relative amounts of time required for each step, and whether a step is sequential with regard to the prior step.

Caution:

To avoid a possible 515106 error (A relation cannot be saved because it violates a unique index), you must translate the TC XML import file when you are exporting from an earlier version of Teamcenter to the current version. If you use the site consolidation default import transfer mode (**SiteConsolidationDefaultTMImport**), Teamcenter translates the import file for you automatically during the import process.

If you use custom transfer modes to import data, you must attach the **Mapping_of_copy_stable_id.xsl** style sheet to the custom transfer mode to translate the import file during the import process. This file is located in the *TC_DATA* directory. Use the **plmxml_tm_edit_xsl** utility to attach the style sheet to the transfer mode.

Note:

For Teamcenter 11.2.3 and later versions, importing data using low-level TC XML with the default **SiteConsolidationDefaultTMImport** transfer mode uses an XSLT file. The XSLT file supports mapping of data from databases that do not support copying a stable ID. The transformation allows importing a relation with the same settings it contains in an upgraded database.

During the import process, the **tcxml_import** utility generates the required **trans.xml** XSL file.

Step	Activity	Tool / comments	Time	Sequential
1a	Replicate	sitcons_replicate_mgr If doing manually: <pre>tcxml_export -low_level tcxml_import -low_level tcxml_confirm_export</pre>	65%	Not Applicable
<div style="border: 1px solid orange; padding: 10px;"> <p>Caution:</p> <p>If cacheless search is used at the source site, you must update the index boxes required by cacheless search at the target site. To update the index boxes, you must have the Jttobboxandtso and Qsearch_process_queue translator services enabled at the target site. Also, after the data is imported, you must run the following commands at the target site:</p> <pre>qsearch_process_queue -u=admin-user -p=password -g=dba -force_queue_all_possible_updates qsearch_process_queue -u=admin-user -p=password -g=dba -process_queue</pre> </div>				
1b	Synchronize	sitcons_replicate_mgr If doing manually: <pre>tcxml_export -sync tcxml_import -low_level tcxml_confirm_export</pre>	5%	Yes
2	Final replicate and synchronize	sitcons_replicate_mgr If doing manually: <pre>tcxml_export -sync tcxml_import -low_level tcxml_confirm_export</pre>	1%	Yes

Step	Activity	Tool / comments	Time	Sequential
		sitcons_user_folders		
3	Final pre-TCP validation	Use validation tools to verify the content in the partition for this increment was brought over to the target site.	-	Yes
3a	Validate	<p>Run the PLM report utilities at the source and target sites using the same input files used for replication. Compare the results to identify any missing or out-of-date data.</p> <p>Replicate any missing or out-of-date data with forced retraversal.</p> <p>Use the sitcons_xfer_owner_mgr utility with the dryrun option and source_extinct reporting options.</p> <ul style="list-style-type: none"> • Compares source site accountability tables to objects in the target site and notes any failed imports. • Reports source site data at the target site that was moved there by methods other than the site consolidation tools. • Reports changes to the replicated islands at the target site not found on the source site. 	5%	Parallel with 3a and 3b
3b	Review and address missing items and objects in validation report.	User reviews and approves to proceed into TCP.	15%	Yes

Replication phase orchestration process

Teamcenter Integration Framework orchestration helps transfer data from the source site to the target site during the *replication phase*. The Teamcenter Integration Framework initiates data replication or synchronization from the source to the target site and provides a scheduling mechanism to allow the option to run a data transfer immediately or process it during off-hours.

A data transfer is composed of the following processing steps:

1. *Schedule* determines when a data transfer should be run.
2. *Export* reads the objects to be exported and writes this to file.
3. (Optional) *Map* maps the exported objects in the file to the schema of the target site.
4. *Import* imports the exported objects.
5. *Confirm export* confirms to source site that objects have been imported.
6. *Notify* emails the data transfer requester of the completion status.

Teamcenter Integration Framework orchestrated replication supports automatically breaking a replication process into parallel processing streams. For transfer actions that are broken into parallel streams, Teamcenter Integration Framework uniquely identifies and monitors every transaction and subtransaction.

Caution:

When using this capability, check the logs of these process steps for rollbacks or failures during the confirm step. An island rollback or object confirmation failure can occur due to too many parallel streams causing lock contention at the database level. These types of errors *do not* cause the replication process or stream to abort. Therefore, normal termination of a process does not imply there were no issues with the load or confirm steps.

Time critical period

Time critical period activities

The *time critical period* (TCP) is the period during which ownership of product data is changed from the source to the target site. During this period, the source and target site are expected to be unavailable to production users. Therefore, this period must be minimized and predictable. The following table summarizes the steps performed and the role of the site consolidation tools in this period.

Step	Activity	Tool / comments	Target site	Source site
1	Back up database.	Perform an Oracle data dump or system snap. Verify Emulation Lab testing is consistent.	10%	10%

Step	Activity	Tool / comments	Target site	Source site
		Can occur after replication phase step 3b.		
2	Synchronize data.	Use the sitcons_replicate_mgr utility to replicate any new data created since the end of the replication period. Use the same utility with the -sync option to ensure any previously replicated data that have been modified are up-to-date. Source site extract can begin immediately upon completion of source site backup.	5%	5%
3	Validate content.	Validate content at the target site before ownership change.	45%	45%
3a	Verify metadata.	Verifies the source site metadata at the target site for objects in partition are up-to-date. Use the plm_report_extract utility to extract, compare, and verify the results of step 2. You can also use the sitcons_xfer_owner_mgr or tcxml_xfer_ownership (low-level) utility with the -dryrun option to determine the objects that will have their ownership changed. For the final execution increment, include the -source_extinct option.	–	–
3d	Review reports.	Review reports and validation information. You need to review only newly synced data. Review and approve other data prior to start of the TCP.	–	–
4	Change ownership.	If using the low-level utility directly, changes ownership at the target site. If using Teamcenter Integration Framework orchestration, changes ownership at target and source sites. Assumes sequential operation, but can be sequential or parallel at the target and source sites. For example, the target site ownership changes can occur before any processing starts at the source.	–	–

Step	Activity	Tool / comments	Target site	Source site
4a	Target site fast ownership change.	<p>If using Teamcenter Integration Framework orchestration, use the sitcons_xfer_owner_mgr utility.</p> <p>If using the low-level utility (tcxml_xfer_ownership) directly, execute the extract and the source site, move the output file to the target site, and invoke the perform action at the target site.</p> <p>Sequential, after TCP step 3d.</p>	25%	–
4b	Source site fast ownership change.	<p>If using Teamcenter Integration Framework orchestration, this action is performed by the sitcons_xfer_owner_mgr utility during step 4a.</p> <p>If using the low-level utility (tcxml_xfer_ownership) directly, execute the ownership change at the source site.</p> <p>Sequential, after TCP step 4a.</p>	–	25%
5	Test sites.	<p>Use the plm_report_extract utility to extract, compare, and validate that the ownership change occurred.</p> <p>Use the rich client, NX, and other user tools for final user testing and other user operations.</p> <p>Can occur simultaneously at the source and target sites.</p>	15	15
6	Open for production.	Occurs after TCP step 5.	–	–

Performance considerations during the time critical period

Site consolidation database transactions typically consume a high amount of database resources as compared to the average daily work load on the Teamcenter production database. On a large database that is tuned for day-to-day operations, site consolidation operations might result in increased input/output (I/O) wait time for database operations. This performance consideration is especially important during the time critical period because the sites are unavailable to the production users for the entire duration of this period. A step-wise tuning process must be instituted to manage the resource demands during this period. The buffer cache, SGA size, and other performance parameters must be tuned appropriately with the focus being on the I/O coming from the large tables and indexes, and the

available memory resources that will be in demand. Once site consolidation (or the time critical period) is complete, the databases may need to be re-tuned to the production workload.

Disabling user access and backing up data

The methodology for disabling user access and backing up data is left up to the user's processes. It is important because this is the recovery point if anything needs to be restored. During this step, the following occurs:

- All production users of the source and target systems are logged off, and the system is inactive with respect to their activities. Access is not reestablished until the end of the time critical period.
- A backup or snapshot of the source and target data base is taken so that, if necessary, it can be restored.

Performing final replication and synchronization

Replication and synchronization enables a final update of new or previously replicated data prior to ownership change. Final replication and synchronization is required to:

- Update the target site objects after the end of the replication phase. There may be production activity after the last replication and synchronization done during the replication phase resulting in modifications to previously replicated or newly created data. Because the sites are inactive, this is the time to completely update the previously replicated data.
- Perform a mass change of ownership and validation before reopening the sites to production operation. This allows the consolidation processes to use as brief a period of downtime as possible to back up sites.

Performing validation

Validation reports support the final audit process. This ensures the desired consolidation information is moved and appropriate ownership changes were made at the target site.

The **tcxml_xfer_ownership** utility, with the **dryrun** and **source_extinct** reporting options, lets you validate information brought over by the final synchronization because this same activity was done at the end of the replication phase to ensure readiness to enter time critical period.

Change object ownership

Changing ownership of data transferred from the source site to the target site is a key activity of the time critical period.

Warning:

To avoid erroneous operations that may result in corrupt data, before using the **tcxml_xfer_ownership** or **sitcons_xfer_owner_mgr** utility to transfer ownership, run the utility with the **-dryrun** argument and verify that replicas exist at the target site for all objects that are having their ownership transferred.

1. Use the **tcxml_xfer_ownership** utility with the **extract** option to extract information from the source site needed to change ownership at the target site.
2. Move the file to the target site.
3. Use the **tcxml_xfer_ownership** utility with the **perform** option to change data ownership at the target site.
4. Move the resulting conformation file to the source site.
5. Use the **tcxml_xfer_ownership** utility with the **update_status** action to perform the ownership change.
6. Run the **sitcons_accountability_chk** utility at the source site to obtain a list of non-consolidated objects for a given class. This produces a report which lists the UIDs of all such objects along with the names of any **Workspaceobjects** class objects for any given class. Review the report to verify that all the required objects have been consolidated to the target site.

Sequencing change ownership tasks automatically

The orchestration framework utility, **sitcons_xfer_owner_mgr**, lets you automatically sequence the tasks involved in ownership transfer.

You use a two-phase strategy to ensure that ownership has been properly changed at the target site prior to resetting the source site ownership.

- Target site evaluation
- Source site evaluation

If the orchestration framework is not used, you must manually sequence ownership change activities at the source and target sites.

5. Cleanup phase activities

Determining when to perform cleanup activities

Some cleanup actions can be performed after each execution increment. You must tailor the use of these activities to the site consolidation plan for your environment.

Resolving issues after moving consolidation information

After all the source site consolidation information is successfully moved, site definition information is compatible between the source and target sites.

The site-to-site sharing situation can be similar to the situation shown in the following table.

Site	Has source site owned replicas	Has target site owned replicas	Has other site owned replicas	Has external site owned replicas
Source site	No	Yes	Yes	Yes
Target site	No	No	Yes	Yes
Other site	Yes	Yes	No	Yes
External site	Yes	Yes	Yes	No

After site consolidation, the product data cleanup requirements include resolving the following issues:

Site	Category of objects
Target	Objects from source site that have been transferred with ownership change and that have import export record (IXR), item export record (ITXR), or publication audit record (PAR) issues.
ODS	Publication record (PR) objects.
Third	Replicas of objects. Objects with IXR, ITXR, or PAR issues.
Source	Objects consolidated to target site with ownership change that also have IXR, ITXR, or PAR issues.

When resolution is deferred from the preparation phase for ownership conflicts other than those between the source and target sites, those conflicts can be resolved in this cleanup phase.

Cleaning up the target site

After the execution phase, the target site:

- Is in production operation.
- Contains all the consolidation information.
- Is equivalent, for production purposes, to the source site.

The target site also has:

- Network connectivity to all its clients including those added as result of the consolidation that meet the assumed criteria.
- A database and volumes not near, in the network sense, the end-user client communities served by the Teamcenter site.
- New or preexisting FSCs near the end-user communities previously served by the source site.
- Site administrative, system, organization, and schema data compatible with the source site.
- Inherited, newly owned data from the source site that is shared with third sites.
- External sites that are accessible for objects from the source site that are transferred with ownership change and that also have IXR, ITXR, or PAR data.
- Correct supporting import/export information for objects from the source site that are transferred with ownership change that also have IXR, ITXR, or PAR data.
- Inherited, from the source site, replica data owned by existing third sites, with the following characteristics:
 - The target site contains data from other and external sites that was replicated to the source site prior to consolidation.
 - Other Teamcenter sites and external sites can be accessed by the target site after consolidation.
 - The supporting import-export information is correct at the target site.

The volumes existed in a *shared* state during consolidation. After consolidation, you may need to decommission the source site or move the volume data.

- To decommission the source site, you must first remove the source site volume entries from the shared FMS primary configuration file. This makes these volumes unavailable to the source site, but leaves them accessible through the target site.

- To move the volume data nearer to the target site, you can use Teamcenter volume migration tools, that move the files over the network, or you can physically relocate and reconnect the volume storage.

Cleaning up the ODS

After completing an increment or after completing all increments, the Object Directory Service (ODS) publication information must be updated for any objects published by the source site prior to consolidation. This must be done for publication record objects before the source site can be decommissioned.

Use the **data_sync** and **data_share** Multi-Site utilities to assist in ODS cleanup.

For example, after objects are moved from the source site to the target site, all ODSs to which the source site has published must be updated because the parts are no longer owned or published by the source site. You can do this by running the **data_sync** utility with the **-republish**, **-update**, and **-force** arguments at the target site, as follows:

```
data_sync -u=tc-admin-user -p=password -g=group -republish -update -force
```

You can also use the **data_share** utility to unpublish the source site objects and then mass publish them from the target site, for example:

```
data_share -u=tc-admin-user -p=password -g=group -f=unpublish
  -filename=source_site_parts.txt -batch_size=16000
  -error_file=massunpub_error.txt -continue_on_error
  -report=massunpub_report.txt -site=ods_site_name
```

```
data_share -u=tc-admin-user -p=password -g=group -f=publish
  -filename=source_site_parts.txt -batch_size=16000
  -error_file=masspublish_error.txt -continue_on_error
  -report=masspub_report.txt -site=ods_site_name
```

Cleaning up third sites

You can use the **data_sync** utility and the site consolidation **sitcons_fix_ixr** utility to resolve inconsistencies with publication records and export records at third sites and to fix information at third sites for shared objects that are consolidated from the source site to target site. This applies to objects that are consolidated to the target site with ownership change that also have IXR, ITXR, PAR, and publication record data to be updated.

- Use the **sitcons_fix_ixr** utility to update publication records and export records for the consolidation data at the third site.
- Use the **data_sync** or **data_share** utility to update the replicas at the third site to show they are replicas from the target site.

Perform third-site cleanup operations for the objects in an execution increment after the increment completes. You must perform these actions for the corresponding objects at the source, target, ODS, and third sites involved in sharing the objects.

- For objects owned by source site moved to the target site during one or more increments of the execution phase of site consolidation, note the following:
 - To update owner information for replicas at third sites, use the **data_sync** or **data_share** utility at the target site.

To use the **data_sync** utility:

```
data_sync -item_id=Eng_1 -force -site=third-site -sync -update
-report=rep.lst
```

To use the **data_share** utility:

```
data_share -item_id=Eng_1 -site=third-site -sync -update -report=rep.lst
```

- To destroy all the export records for a given site and replace the references to the extinct site with references to the replacement site (**Site1**), use the following command:

```
data_sync -site=DeadSite -verify -assert_extinct_site -replacement_site=Site1
```

The previous command makes **Site1** the owning site of all the replicas owned by **DeadSite** at the third site.

Note:

The **-replacement_site** argument is valid only with the **-assert_extinct_site** argument and identifies the site that replaces the site to that is being removed.

For objects published to the ODS from the source site, the publication access records (PARs) at the ODS assume the source site as the owner of the object. **Repair the ODS** published objects to correct this consolidated objects.

- The IXR, ITXR, and PAR of the objects earlier owned by source site and moved to target site still exist at the source site. **Repair the source site records** by removing these objects at the source site. This step is not required if you plan to retire the source site.
- For replica objects owned by the third site that are moved from the source site to target site, note the following issues:
 - The IXR, ITXR, and PAR data at the third site lists objects as exported to the source site. You must update the target site properties of the IXR, ITXR, and PAR data attached to the owned object at the third site.

```
sitcons_fix_ixr -u=tc-admin-user -p=password -g=group -f=fix_records
-source_site=source-site-name -target_site=target-site-name
```

- For multiple sites that own the replica objects moved from the source site to the target site as part of the consolidation, the export records for these objects at each of the owning third sites must refer to the target site. Run the **sitcons_fix_ixr** utility at each of the third sites for the given set of replica objects moved from the source to the target site that are owned by the third site as previously described.

Cleaning up the source site

After the target site, third site, and ODS cleanup, the source site can be retired or, optionally, it can be left online and available to system administrators to address issues at the remaining sites that continue in the production operation.

If a source site is to be operational after the ODS and third sites are cleaned up and source site references in the data are moved to the target site, you must clean up the source site. This becomes an issue if you perform cleanup after an execution increment completes but before the final execution phase increment.

At the source site, you must remove the IXR, ITXR, and PAR data of the objects formerly owned by the source site that have moved to the target site. Use the **sitcons_fix_ixr** utility with the list of moved object UIDs. For each object in the input list, the utility deletes the IXR, ITXR, and PAR data attached to the object.

If performing site consolidation and the source site is used to clone the data to another site, clean the source site records with **app_id =':SITECON_SYNC'** in **acct_table** using the following SQL:

```
delete from acct_table where app_id =':SITECON_SYNC'
```

Determining final cleanup steps

After you finish target site, ODS, and third site cleanup, you must determine whether to retire the source site or leave it operational. You can retire the source site when all access requirements are removed and the client base is using the consolidated target site. If you decide to leave the source site operational, you must **clean up the site**.

You must also determine where to locate the source site volume data.

- You can leave your data at its current location if, for example, you have NX users at that site and the local volume provides access to data that is generally required for the local users. Storing the NX files locally improves data access and reduces the need to pull the files across the network.
- You can move volumes to a target data center using either of these options:
 - Use volume management tools to migrate volume data at the source site to the target site.

- Relocate physical storage hardware, such as the servers and disks, to the target site. Incorporating the volumes into the target site provides the resource for volume data.

When all the volume data is configured at the target site, you can update the volume path in the primary FMS configuration file to remove volume information that references the source site.

A. Site consolidation utilities

admin_data_export

Exports administration data to a file. You can use the file to import the data to another site, such as from a test or upgrade environment into a production environment. You can also use the file to compare data between two sites. You can alternatively perform full or partial export of administration data using TEM.

SYNTAX

```
admin_data_export -u=user-ID {-p=password | -pf=password-file} -g=group
{-adminDataTypes=data-type1, data-type2,...,data-typeN | all}
{-outputPackage=absolute-path-file-name}
[-inputCriteria= <className>{attr1=value1,attr2=value2,...,attrN=valueN}]
[-listTypes]
[-session_options=option1:value1,option2:value2:...]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

If this argument is used without a value, the operating system user name is used.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-ID* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-ID* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-adminDataTypes

Specifies a comma-separated list of administration data types you want to export. You can also specify the **all** value to export all the administration data types from the local site.

-outputPackage

Specifies the fully qualified path to the location for the generated export file.

-inputCriteria

Specifies the criteria used to generate an export file containing a partial amount of the site's administration data. You must specify the class name of the administration data first, followed by the attributes and their values that identify the data you want to export.

If you do not specify this argument, the utility does a full export of all administration data at the site. Following are some examples of input criteria. The attributes and values listed here may not match your Teamcenter data.

Admin data type	Selection by	Class name	Example attribute
AccessManager	Access Control List	AM_ACL	{ACL_Name=value}
LogicalObjects	Logical Object Name	LogicalObjectsByName	{type_name=value1,type_name=value2}
Organization	User	User	{user_name=value}
	Group	Group	{name=value}
Projects	Projects	TC_Project	{project_id=value}
Preferences	Preference Name	PreferenceByName	{preference_name=value}
	Group	PreferencesByGroup	{group_name=value}
	Role	PreferencesByRole	{role_name=value}
	User	PreferencesByUser	{user_name=value}
	Category	PreferencesByCategory	{category_name=value}
RevisionRules	Rule	RevisionRules	{object_name=Latest*}
SavedQueries	Query Name	ImanQuery	{query_name=value}
Stylesheets	Stylesheet Name	StylesheetByDatasetName	{dataset_name=value}
	Preference Name	StylesheetByPreferenceName	{preference_name=value}
Subscriptions	Subscription	ImanSubscription	{fnd0Name=val}

Admin data type	Selection by	Class name	Example attribute
UnitDefinitions	ID	UnitDefinitionsById	{unit_definition_id=value}
WorkflowTemplates	Template Name	WorkflowTemplatesByName	{template_name=value}

-listTypes

Displays a list of administration data types that can be exported.

-session_options

Indicates session options that you want the utility to use for the current export. Accepts valid session option-value pairs in a comma delimited format. The option name and value are delimited with a colon.

Caution:

If you do not supply this argument, the **opt_traverse_ref_org** option value is set to **True** and the utility skips user information updates.

You must export or import organization objects (such as users and groups) before importing other objects. Otherwise, the import of other objects fails.

If you supply the **opt_traverse_ref_org** option with the value set to **False**, the utility updates user object data. This may cause the user password to change at the importing site.

-h

Displays the help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

This utility does not function if the ownership of 4GD replica objects is transferred to the target site. It must be run before any ownership transfers take place.

EXAMPLES

- To export all organization, preferences, and projects administration data from a site:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -adminDataTypes=Organization,Preferences,Projects
  -outputPackage=c:\temp\admin_data\siteA\siteA.zip
```

- To list the available administration data types:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -listTypes
```

- To do a partial export of only Access Manager rules named **System**:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -adminDataTypes=AccessManager
  -inputCriteria=AM_ACL{ACL_Name=System}
  -outputPackage=c:\temp\admin_data\export_partial_am_system.zip
```

- To do a partial export of only Workflow Templates named **TCM Release Process**:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -adminDataTypes=WorkflowTemplates
  -inputCriteria="WorkflowTemplatesByName{template_name=TCM Release Process}"
  -outputPackage=c:\temp\admin_data\export_partial_wftemplate_release.zip
```

- To export Access Manager rules with referenced organization data:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -adminDataTypes=AccessManager
  -session_options=opt_traverse_ref_org:false
  -outputPackage=c:\temp\admin_data\export_partial_am_system.zip
```

- To perform a partial export of organization data, pass the input criteria and session options, with the **-opt_traverse_ref_org** option value set to **False**.

- **Example 1**

To export the user whose user name is **PLMUser**:

```
admin_data_export -u=admin-username -p=admin-password -g=dba
  -adminDataTypes=Organization
  -outputPackage=C:\Temp\OrganizationPR_User.zip
  -inputCriteria=User{user_name=PLMUser}
  -session_options=opt_traverse_ref_org:false
```

- **Example 2**

To export the group having the name **PLM**:

```
admin_data_export -u=admin-username -p=admin-password -g=dba  
-adminDataTypes=Organization  
-outputPackage=C:\Temp\OrganizationPR_Group.zip  
-inputCriteria=Group{name=PLM}  
-session_options=opt_traverse_ref_org:false
```

admin_data_import

Imports administration data from the specified input package into the current Teamcenter environment. (To create a site package file, export administration data.)

A conflict occurs when the data exists at both the exporting and importing sites but does not contain the same values at both. For each administration data type, you can specify how conflicting data is handled during import.

The utility allows you to generate an import report without executing the actual import, referred to as a dry run. You can use the import report (dry run) to analyze how the input package potentially impacts the current environment. For an actual import, the utility adds a report to an import history report that contains a record of all the previous imports performed at the current environment. You can use this report to analyze how administrative data has changed over time at the importing site due to prior imports. Import reports (dry run and actual) are located in the `TC_ROOT\logs\import_history` directory.

Caution:

If you have an input file with Access Manager data from older Teamcenter releases (earlier than Teamcenter 13.1), ensure that the input file is regenerated with a minimum Teamcenter 13.1 version before importing.

SYNTAX

```
admin_data_import -u=user-ID {-p=password | -pf=password-file} -g=group
-inputPackage=input-file-name-and-path
-adminDataTypes=admin-data-type1,admin-data-type2, ... | all
-mergeOption=admin-data-type1,merge-option1:admin-data-type2,merge-option2, ...
[-dryrun] [-skipPackageValidation] [-listTypes] [-listMergeOptions]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-inputPackage

Specifies the fully qualified path to the location for the file to import.

-adminDataTypes

Specifies a comma separated list of administration data types you want to import. You can also specify the **all** value to import all data types in the input package.

Tip:

Use the **-listTypes** argument to display a list of allowable administration data types.

-mergeOption

Indicates how the utility handles data conflicts (merge options) between the target (importing) site administration data and the source (exporting) site data in the input package.

Tip:

Use the **-listMergeOptions** argument to display a list of allowable merge options for each administration data type the import file contains.

-dryrun

Generates an import history report that shows the impact on the importing site's administration data without actually importing the data.

-skipPackageValidation

Specifies the import file is not validated for proper content prior to importing the package.

If you do not specify this argument, the utility ensures the data in the import file is valid and fails the import if it is not.

-listTypes

Displays the supported administration data types that can be imported from the input package.

-listMergeOptions

Lists the supported merge options configured for the available administration data types in the import file. You must specify the **-inputPackage** argument to point to the import file.

The merge options are:

override_with_source

Replaces the target site administration data with the source site administration data.

keep_target

Keeps the existing administration data at the importing site when there are merge conflicts.

choose_latest

Replaces the conflicting administration data with the latest administration data based on time stamp.

choose_target_for_unresolvable_conflicts

Keeps the existing administration data at the importing site for nontrivial or unresolvable merge conflicts. For all other merge conflicts, uses the latest administration data based on time stamp.

session_options

Indicates session options that you want the utility to use for the current command. Accepts valid session options in a comma delimited format.

If you supply the **opt_traverse_ref_org** in this argument value, the utility traverses the organization data. Otherwise its value is passed as true (default behavior).

Caution:

You must export/import organization objects (such as users and groups) before importing other objects. Otherwise the import of other objects fails.

-h

Displays the help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment.

FILES

As specified in Log files produced by Teamcenter.

EXAMPLES

- Imports administration data from the specified input package into the current Teamcenter environment:

```
admin_data_import -u=admin-username -p=admin-password -g=dba
-adminDataTypes=all -inputPackage=C:\temp\admin_data\siteB\siteB.zip
```

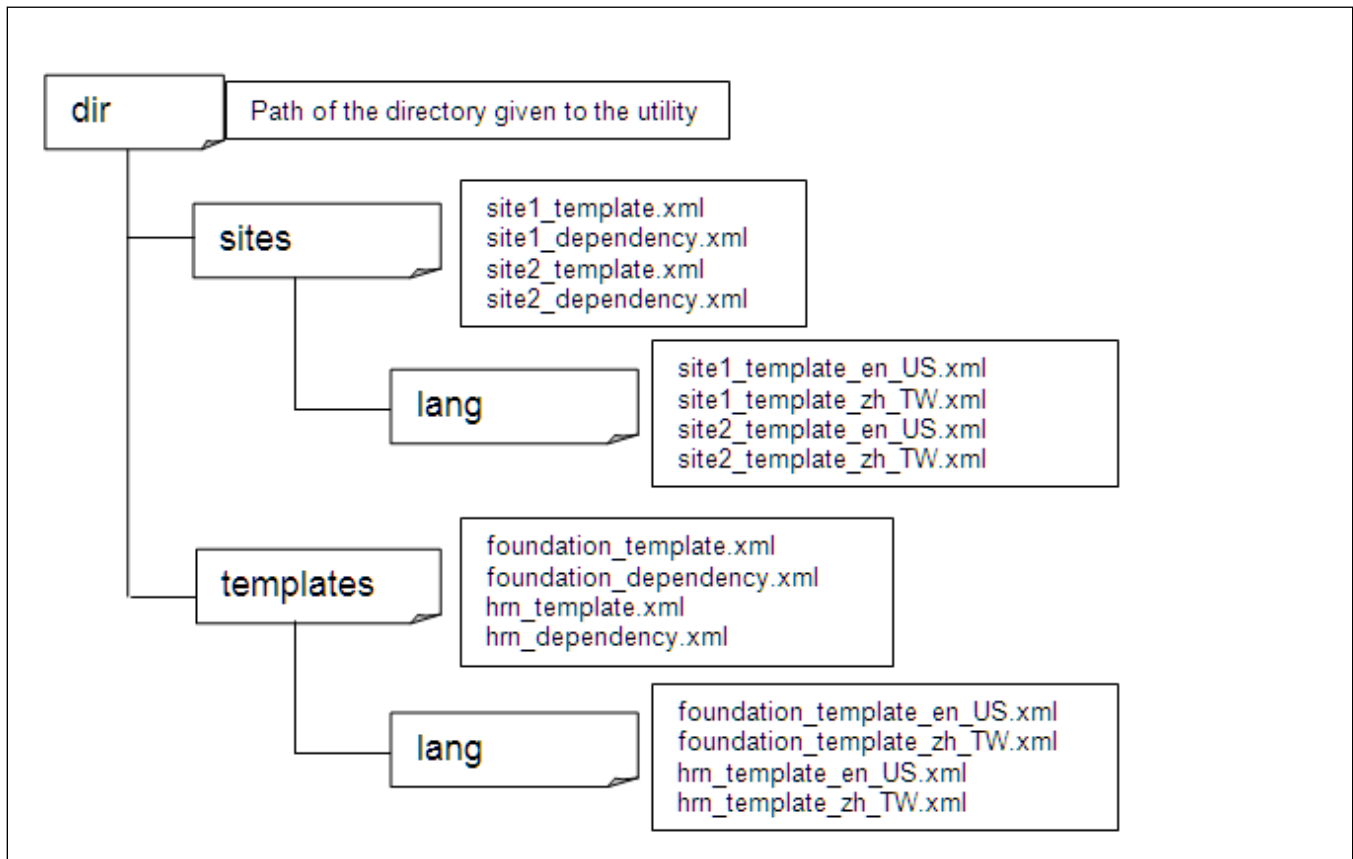
- Performs a dry run import from the specified input package with merge options:

```
admin_data_import -u=admin-username -p=admin-password -g=dba
-dryrun -adminDataTypes=all
-inputPackage=C:\temp\admin_data\siteB\siteB.zip
-mergeOption=AccessManager:override_with_source,Organization:
override_with_source,Preferences:choose_latest,
RevisionRules:choose_target_for_unresolvable_conflicts,SavedQueries:keep_target,
Stylesheets:keep_target,
Subscriptions:keep_target,WorkflowTemplates:override_with_source
```

bmidе_commontemplategenerator

Generates a common template between two or more sites and generates site-specific templates for each site. Use this utility when you have multiple sites containing custom data model. Previously, you had to manually analyze the site templates generated at each site to create a common template. This utility eliminates the manual work to create a template that contains custom data model common to all sites.

Place template files in the following directory structure:



Directory structure required by the bmidе_commontemplategenerator utility

Obtain the files to place into the directories from the packaged *template-name_template.zip* files. The **sites** directory contains the templates from the different sites, the **templates** directory contains the standard templates that the site templates are dependent upon, and the **lang** directories contain the localization files used by the templates.

SYNTAX

bmidе_commontemplategenerator

-dir=*site-templates-directory*
 -name=*common-template-name*
 -displayname=*common-template-display-name*

-outputdir=*output-directory*
[-h]

ARGUMENTS

-dir

Specifies the path of the directory where sites and templates folders are located. The directory should contain a **sites** subdirectory that contains all the site templates and dependency files and a **templates** subdirectory that contains all the dependent templates. The **sites** and **templates** directories should contain a **lang** subdirectory containing locale files.

-name

Specifies the name of the common template to be generated.

-displayname

Specifies the display name of the common template to be generated.

-outputdir

Specifies the path of the directory where the generated files are to be placed.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

This utility has a requirement that a minimum of 1500 MB memory must be allocated to the Java heap space. Before running the utility, set the following environment variable:

```
set BMIDE_SCRIPT_ARGS=-Xmx1500M
```

EXAMPLES

```
bmide_commontemplategenerator -dir=c:\templates\temp  
-name=commontemplate -displayname="Common Template"  
-outputdir=c:\templates\temp\output
```

bmide_comparator

Compares two complete Teamcenter model files and generates a differences file. This utility must be run with either the **-schema** or **-all** argument.

SYNTAX

```
bmide_comparator -compare={schema | all} -old=old-model-file-path  
-new=new-model-file-path -delta=differences-file-path  
[-log=log-file-path] [-h]
```

ARGUMENTS

-compare={schema | all}

Compare data model. You must specify one of these options:

- **schema**

Compares only classes.

- **all**

Compares all elements.

-old

Specifies the file path and name of the file containing the old Teamcenter model.

-new

Specifies the file path and name of the file containing the new Teamcenter model.

-delta

Specifies the file path and name of the file into which data model differences will be written.

-log

Specifies file path and name of the log file that contains the results of this execution. This argument is optional.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

database_verify

Compares database schema, Teamcenter types, tools, release statuses, and units of measure between two specified Multi-Site Collaboration sites and generates a report of any database discrepancies.

You can use this utility to query types and classes from a specified remote site and create a local dataset named **TCTYPES_SITE***siteid* containing those mappings. There is one dataset for each remote site defined with the **-site** argument. You can also create and update the local dataset of remote class and types mappings for a specified remote site. Run this utility whenever there are changes for the POM transmit file of a specified site.

SYNTAX

```
database_verify [-u=user-id {-p=password | -pf=password-file} -g=group]
  -from=site-name1 -to=site-name2
  [-schema] [-type] [-tool] [-status] [-uom] [-all] [-output=file-name]
  [-site=site-name] -force -offline [-filename=file-name] [-v] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-from

Specifies a Teamcenter database site to be verified.

-to

Specifies a Teamcenter database site to be verified.

-output

Specifies the output format. The report is output to a file if a file name is specified. If not, the report is displayed in a shell.

-schema

Compares schema between the two sites.

-type

Compares types between the two sites.

-tool

Compares tools between the two sites.

-status

Compares release status types between the two sites.

-uom

Compares units of measure between the two sites.

-notetype

Compares note types.

-all

Compares classes, types, tools, status types and units of measures between the two sites. This is the default if no argument is supplied.

-site

Specifies the site name where types and classes would be persisted locally. If the value of this argument is set to **ALL**, the utility generates these datasets for all sites in the database.

-force

Generates the type-class mappings file even when the POM transmit files for the remote and local sites have not changed.

-offline

Specifies the site identified with the **-site** argument is offline. If you specify this argument, you must also specify the **-filename** argument.

-filename

Specifies the file name generated by this utility from the **-site** argument. This argument is required if the **-offline** argument is specified.

-v

Runs utility in verbose mode. Displays maximum amount of information. Typically, nonverbose utility sessions only display error messages.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

- To use this utility, you must be a user with system administration privileges or be granted authorization by a user with system administration privileges.
- The **-from** and **-to** arguments must be specified.

data_share

Used for various Multi-Site Collaboration operations, such as publishing and unpublishing objects collectively and sending objects to remote sites. It can be used as a deployment tool during the initial Multi-Site Collaboration implementation phase or as a day-to-day tool for performing functions that previously were available only through the user interface. This utility is especially helpful in setting up and maintaining a hub configuration.

The behavior of this utility is controlled by the **TC_force_remote_sites_exclude_files** preference. If this preference is set to **true**, the replica files are stored in the remote site FMS server cache (FSC); otherwise, the replica files are stored in the remote system volume.

The behavior of the utility for project relationships on replica objects can be controlled by the **TC_sync_projects_with_owning_site** preference. This preference is not created by the Teamcenter installation process. To change the default behavior of shared project relationships, you must create the preference.

The utility also supports TC XML transfers for of 4th Generation Design (4GD) data. The 4GD relation data mapping is controlled by the **TC_cms_relation_optset_map** preference. Use this preference when you want to control the relations that are included or excluded when replicating a 4GD object.

This utility supports part family templates and part family members. Use this utility to:

- Mass publish objects to one or more Object Directory Services (ODS) sites.
- Mass unpublish objects from one or more ODS sites.
- Publish or unpublish an entire assembly.
- List ODS sites currently defined in the local database and authorized for publication.
- Send objects to other sites.
- Delete obsolete publication records at the ODS.
- Check current status of authorized publication sites.
- List ODS sites to which an object is published.
- Import an item from a remote site.
- Export 4GD data in TC XML format.

Data can be input to this utility in the following forms:

- Input file

- Folder name
- Object ID template

When sending objects to a specific user and/or group at a remote site using the **-owning_user** and **-owning_group** arguments, the following rules apply:

- If both the specified user and group exist at the importing site, the imported objects are owned by the user and group regardless of whether or not the user is a member of the group.
- If only the user is specified or if the group is specified but does not exist at the importing site, the user's default group at the importing site is the owning group of the imported objects.
- If only the group is specified or if the user is specified but does not exist at the importing site, the user context of the remote IDSM process is the owning user of the imported objects.

AVOIDING OUT-OF-MEMORY ERRORS

To avoid possible out-of-memory errors when you are replicating architecture revisions, Siemens Digital Industries Software recommends you exclude **MEAppearancePathNode** (APN) objects by setting the following environment variable:

```
TC_EXCLUDE_APN=TRUE
```

This excludes APN objects from the export or import. It may also exclude associated object, such as JT for promoted and deformed bodies. You can replicate the APN objects using the **sync_product_apns** utility.

If you do not set this environment variable, or you set its value to **FALSE**, you can use the batch feature of this utility to overcome memory limitations when replicating large numbers of APNs. For example, you may specify:

```
-batch_objects=MEAppearancePathNode -batch_size=5000
```

When setting the batch size, consider the number of APNs generated by your business processes and the amount of memory available on your system.

SYNTAX

```
data_share [-u=user-id {-p=password | -pf=password-file} -g=group]
-f={function} [-site=remote-site-name1 -site=remote-site-name2... ]
[-owning_user=remote-user] [-owning_group=remote-group]
{-item_id={item-id | template} | -folder=folder-name |
-name=workspace-object | -filename=input-file |
[-key=keyAttr1=keyVal1,keyAttr2=keyVal2...,keyAttrN=keyValN |
-itemKeyFile=file-name] | [-itemRevisionKeysFile=file-name]}
[-class=wso-class-name | -classoffile=class-name]
[-include=relation-type1 -include=relation-type2...]
```

```

[-exclude=relation-type1 -exclude=relation-type2...]
[-revision-selector | -rev=rev-id ]
[-include_bom] [-include_modified_only]
[-assert_precise] [-transfer] [-attach]
[-exclude_files] [-latest_ds_version] [-exclude_folder_contents]
[-include_bc] [-include_supercedures] [-include_pfmembers]
[-include_pftemplates] [-pf_bom_treatment=option]
[-qry_name=query-name -qry_attr=attr-name1 -qry_val=attr-value1[-qry_attr=attr-name2
-qry_val=attr-value2...]]
[-oat] [-continue_on_error] [batch_size=number-objects-per-batch]
[-report=report-file-name] [-user=user-id] [-group=group-name]
[-error_file=error-file-name] [-exclude_variant_options] [-dir]
[-mapping_rule=rule-file-name]
[-batch_variant_options] [-batch_objects=class-for-deferred-objects]
[-batch_file=file-name-for-deferred-objects]
[-include_dist_comp] [-lightweightobject] [-log]
[-checkpoint [-compress_ind_files=S | I | N ] [-transaction_id=transaction-id]
[-tcx_data_load [-migration_sync]]
[-status]
[-cleanup_transaction [-transaction_id=transaction-id | -before_last_process_date=date] ]
[-restart] [-commit_ixr] [-list_transactions]
[-optionset | -optionset=optionset-name [-de_incl_rlz_bom]
[-workset_include_relz_de] [-4gd_id=object-id -class=4gd-class-name] ]
[-all_roles] [-all_subgroups] [-all_groupmembers]
[-include_4gd_baseline_content]
[-parallelize=owner-processes,remote-processes]
[-override_options=option1:value1,option2:value2, ...]
[-session_options=option1:value1,option2:value2, ...]
[-volume_file_list=volume_list]
[-h]

```

ARGUMENTS

Entries in parentheses are accepted abbreviations for arguments.

-u

Specifies the user ID. This is generally a user with administration privileges. Be aware that when **data_share** is run by a system administrator, **Has Bypass ACL** is set and access control list rules are bypassed.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

Caution:

For HTTP enabled sites, remote site operations log on using the default group for the user supplied with the **-u** argument. Any value supplied with the **-g** argument is ignored.

-f

Specifies the function to be performed; you must specify one of the following:

send

Sends objects to the specified remote sites.

The objects to send are determined by the **-item_id**, **-folder**, **-filename**, **-4gd_id**, **-all_subgroups**, **-all_roles**, or **-all_groupmembers** arguments.

publish (pub)

Publishes objects to the given ODS sites. The objects to publish are determined by the **-item_id**, **-folder**, or **-filename** arguments. Cannot be used with the **-all_subgroups**, **-all_roles**, or **-all_groupmembers** arguments.

unpublish (unp)

Unpublishes objects from the given ODS sites. The objects to unpublish are determined by the **-item_id**, **-folder** or **-filename** arguments. Cannot be used with the **-all_subgroups**, **-all_roles**, or **-all_groupmembers** arguments.

delete_pubrec (dpr)

Deletes obsolete publication records for the specified object from the local database. This must be run at the ODS site containing the publication record to be deleted. Only privileged users may use this function. Requires the **-item_id** argument with specific item ID; no wildcards or other arguments are supported with this function.

Note:

To be used only if the primary object has been deleted but publication records still exist at the ODS site.

register (reg)

Registers item IDs to the central item ID registry.

unregister (unreg)

Unregisters item IDs from the central item ID registry. The register and unregister functions must be supplied with the **-item_id** or **-filename** argument. To unregister deleted items, you must run this utility at the central item ID registry site.

delete_exprec (dxr)

Deletes export records for the specified sites for objects listed in the text file identified by the **-filename** and **-classoffile** arguments. It does not traverse item structure. Only privileged users may use this function.

Note:

To be used only as a last resort after attempting to delete export records using the **-verify** argument of the **data_sync** utility.

list_ods (lo)

Lists the authorized ODS sites, which consist of the default ODS site and the sites specified by the **ODS_publication_sites** preference.

check_ods (co)

Lists the availability of authorized ODS sites.

list_pub_info (lpi)

Lists publication information about objects. Must be run at the owning site.

find_duplicates (fd)

Compares all of the item IDs at the remote site specified by the **-site** argument. The item IDs searched for may be filtered with the **-item_id**, **-created_before**, and **-created_after** arguments. The output may be directed to a file using the **-report** argument. The output is formatted to **csv** style, using comma-separated values.

-created_before

Restricts searches for duplicate items to those created at the target site before the specified date.

-created_after

Restricts searches for duplicate items to those created at the target site after a specified date.

remote_checkout

Check out replica objects on a remote site, for example,

```
data_share -f=remote_checkout -filename=uid.txt -
classoffile=Tagstring
```

checkin_replica_co

Check in replica objects on a remote site. Specify the object to check in with **-rep_co_tag**. Must be used with **-session_options** as shown in the following example.

```
data_share -f=checkin_replica_co -rep_co_tag=obj_uid -
session_options=debug:true
```

list_remote_co (lremco)

Lists primary objects that are checked out by remote users based on the specified user ID, group name, and site name.

If no user, group, or site is specified, all remote checkouts are listed.

list_replica_co (lrepc)

Lists replica objects that are checked out from a remote site based on the specified user ID, group name, and site name.

If no user, group, or site is specified, all replica checkouts are listed.

cancel_remote_co (cremco)

Cancels all remote checkouts based on the specified user ID, group name, and site name.

If no user, group, or site is specified, all remote checkouts are canceled.

Note:

Use this argument at the owning site only.

cancel_replica_co (crepc)

Cancels replica checkouts based on the specified user ID, group name, and site name. Canceling a replica checkout also cancels the remote checkout at the owning site.

If no user, group, or site is specified, all replica checkouts are canceled.

Note:

Use this argument at the owning site only.

remote_import (ri)

Imports the item specified by the **-item_id** argument from the owning site or the site specified by the **-site** argument. If the item is a replica at the local site, it is imported from the owning site and any site specified in the command is ignored.

When you specify the **-optionset** argument, Multi-Site uses a TC XML payload to exchange data. For 4GD data, you must specify the **-optionset** argument.

Note:

Wildcard characters cannot be used in the **-item_id** argument.

You can also use this argument to import a list of items from an input file designated by the **-filename** argument. The input file must contain UIDs for the items to be imported, and the **-classoffile** argument value must be set to **Tagstring** when using an input file. You can use the **sync_on_demand** utility to generate a file that contains UIDs of items of an assembly enclosed within square brackets ([]) or other designated separator. You can then write a script to collect the UIDs into the input file.

The **data_share** arguments related to variants and line of usage (LOU) cannot be used with the **remote_import** argument.

By default, items are processed sequentially. Use **-bulk** with **-f=remote_import** to process the items in a single operation.

offline_export

Exports the objects to the directory on the file system specified by **-dir**.

-meta_file

Generates only metadata files without exporting objects. Used only with **offline_export** and **offline_import**. Typically used for data migration.

offline_import

Used with **-dir**, imports the objects to the directory on the file system.

-bypass_site_check

Import metadata file to a site other than the original target site. Used only with **offline_import**.

-meta_file

Generates only metadata files without importing objects. Used only with **offline_export** and **offline_import**. Typically used for data migration.

register_migration_data

Registers the Teamcenter objects to be migrated between different Teamcenter releases. See Migrating Teamcenter data to a newer version of Teamcenter.

install_http_proxy_user

Installs a proxy user at the site specified with **-site** or update the details of the proxy user at the site specified with **-site**. With **install_http_proxy_user**, use the following arguments to specify the proxy user details:

-proxy_user=proxy_user_name

-proxy_group=proxy_user_group

-proxy_role=proxy_user_role

-proxy_user_pwd=proxy_user_password

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

remove_http_proxy_user

Uninstalls the proxy user at the site specified with **-site**. See Configure Multi-Site authentication using secondary LDAP servers configured with TcSS.

-dir

Specifies the local directory in or from which objects are imported or exported. Required with **-f=offline_export** and **-f=offline_import**.

-site

Specifies the name of the site to which objects are published or from which they are unpublished. It can be given multiple times in a command line.

-owning_user (ou)

Specifies the user ID of the user at the remote sites to which the objects are sent. The specified user owns the objects being sent. See [Restrictions](#).

-owning_group (og)

Specifies the name of the group at the remote sites to which the objects are sent. The group owns the objects being sent. See [Restrictions](#).

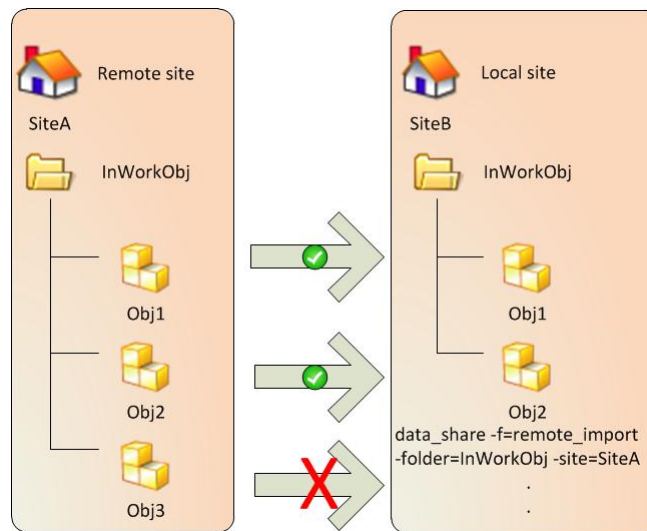
-item_id (item)

Specifies the item ID or template of items to process. It is mutually exclusive with the **-folder**, **-filename**, **-keyFileName**, **-name** and **-key** arguments. It is required for the **-delete_pubrec** argument.

-folder (fl)

Specifies the name of a Teamcenter folder containing the list of objects to process. It is mutually exclusive with the **-name**, **-filename**, and **-item_id** arguments.

During a remote import, if the named folder does not exist at the replica site, the utility generates an error message. If the folder is empty at the replica site, the utility does not pull any objects from the remote site. If the folder exists and has content, the objects that it contains are imported from the remote site.



Note:

If the **-include_bom** argument is used with the **-folder** argument, only the **ItemRevision** objects (and objects related to them) in the folder are replicated. The folder itself is not replicated.

-name

Specifies the name of a single workspace object to be processed. If not an item, use the **-class** argument to specify the class of the object. It is mutually exclusive with the **-folder**, **-filename**, and **-item_id** arguments.

For organization objects, this argument accepts the following attributes:

- **User** objects require a **user_ID** attribute.
- **Role** objects require a **role_name** attribute.
- **Group** objects require the groups full name that uniquely identifies the group.

- **Person** objects require a **user_name** attribute.

Note:

You can use a text file containing a list of all organization objects of the same type that you want to export using the **-classoffile** and **-filename** arguments.

-filename (fn)

Specifies the name of the input file containing the list of IDs or names of objects to process. File entries are treated as IDs for **Items** and **ItemRevisions** objects and as names for other classes of objects. It is mutually exclusive with the **-name**, **-folder**, **-item_id**, **-key**, **-itemKeysFile**, and **-itemRevisionKeysFile** arguments.

If the input file contains names, the **-classoffile** argument is required.

-key

Specifies the keys of the items to process, the template of the item keys, or the 4GD object key. It is mutually exclusive with the **-item_id** argument. Use the following format:

```
keyAttr1=keyVal1,keyAttr2=keyVal2...,keyAttrN=keyValN
```

To find the key of an object, use the **get_key_string** utility.

-itemKeyFile

Specifies the name of the input file containing the list of item key strings of the items you want to update. The following listing shows sample content of a file for updating a list of items:

```
item_id=export_001
item_id=M2Item1_001,object_name=M2Item_name1,object_type=M2Item1
```

If the item key file has 4GD object key strings, the corresponding 4GD class must be supplied using the **-classoffile(cof)=** argument. The following listing shows sample content of a file for updating 4GD data:

```
4gd_id=DE_Export_001
4gd_id=DE00001_ID,object_name=DE00001_Name,object_desc=DE00001_Desc
```

The **4g_id** entry maps to the corresponding unique ID of the 4GD class, for example:

```
Class=Cpd0DesignElement, 4gd_id=cpd0_design_element_id
```

-itemRevisionKeysFile

Specifies the name of the file containing the keys of the item revisions to process. It is mutually exclusive with the **-item_id** argument.

-class (cl)

Specifies the Teamcenter class of the object specified by the **-name**, **-key**, or **-4gd_id** argument. This argument is valid only with the **-name** or, when the **-optionset** argument is specified, with the **-4gd_id** argument. The default class is **Item**.

For organization objects, this argument accepts **Role**, **User**, **Group**, and **Person** classes.

-classoffile (cof)

Specifies the class of the objects listed in the input text file given with the **-filename** argument. If not defined, the default class is **Item**. It is required if the input file has names instead of IDs.

For organization objects, this argument accepts **Role**, **User**, **Group**, and **Person** classes.

-include

Specifies a relation type to include. This can be specified multiple times in a command line. The database name (not the display name) of the relation type must be used.

For best TC XML performance, use a closure rule for relation instead of include/exclude.

-exclude

Specifies a relation type to be excluded from the operation. This can be specified multiple times in a command line. The database name (not the display name) of the relation type must be used.

The list of relation types to be included is determined by either the **TC_relation_required_on_export** (export without transferring ownership) or **TC_relation_required_on_transfer** (export with transfer of ownership) preferences.

The **IMAN_master_form** relation cannot be used as an argument for the **-exclude** argument. The following relations cannot be used as an argument for the **-exclude** argument unless they are not included as a value in the preference for the location:

- **IMAN_requirement**
- **IMAN_specification**

For best TC XML performance, use a closure rule for relation instead of include/exclude.

-revision-selector

Identifies the item revisions to send. It is also used as the revision rule for identifying components when processing assemblies. When used with the **-include_bom** argument while publishing or unpublishing, it determines which revisions' BVR to follow in traversing the assembly tree.

If no revision selector is specified, the default selector is **all_revisions**.

The valid revision selectors are as follows:

all_revisions

Sends all revisions. It is not valid when publishing.

latest_revision

Processes only the latest revision regardless of release status. This is the default if no revision selector is given when publishing or unpublishing.

selected_revision

Process only the selected revision.

latest_working

Processes only the latest working revision.

latest_released

Processes only the latest released revision with any release status.

latest_working_or_any

Sends only the latest working revision. If none, the latest released revision is processed.

release_status

Processes only the latest released revision with the given release status.

all_released_revs

Sends all revisions with a release status; it is not valid when publishing.

-rev

Specifies the ID of a specific item revision to be sent to a remote site. It is valid only with the **-item_id** argument and with the **-send** function, and is mutually exclusive with revision selectors.

You can use the **-rev** and **-include_bom** arguments together to send a precise assembly by including the **-assert_precise** argument.

For 4GD content, you can set this argument as **-rev=baseline_rev** along with the following arguments set as **-4gd=baseline_id** and **-class=MdloBaseline** to select specific 4GD baseline content.

-include_bom (bom)

Includes assembly components when sending, publishing, or unpublishing. A revision selector is required when publishing or unpublishing an assembly; if no revision selector is given, the **latest_revision** selector is used as the default. When sending, the default selector is **all_revisions**.

If not specified, this argument defaults to **off**.

This argument does not traverse the component relationships of subassemblies. This allows you to send the subassemblies in separate transactions. Using multiple, simultaneous transactions to transfer very large assemblies and their subassemblies separately provides improved scalability and performance.

Note:

This argument, although similar to the rich client **Include Entire BOM** remote export option, may not export the same set of objects. The **Include Entire BOM** option traverses all components, subassemblies, and subassembly component relationships that can result in unacceptable performance for very large assemblies.

-include_modified_only

Includes only modified item objects, resulting in improved performance. For example, if an item has revisions A and B, where B is the new revision, only revision B and its dataset are sent to the remote site.

-assert_precise (ap)

Specifies that the assembly is precise. The **-rev** and **-include_bom** arguments can be used together for precise assemblies only. You must include this argument to send a specific revision of an assembly to a remote site.

-transfer (tf)

Transfers site ownership when sending objects. Site ownership is not transferred by default.

-attach (att)

Attaches an object to the appropriate parent item or revision at the receiving site when sending an attachment with transfer of site ownership. Use this for situations in which you attach a dataset to a replica, such as a JT file, and you want to send the JT file to the owning site with transfer of site ownership and attached to the appropriate parent item or revision.

-exclude_files (exf)

Excludes dataset files.

-latest_ds_version (ldv)

Sends only the latest dataset version. Unless this argument is specified, all dataset versions are sent.

-exclude_folder_contents (efc)

Excludes the contents of folders being exported.

-include_bc (ibc)

Identifies the **BomChange** objects associated with the affected assemblies to send. If not specified, **BomChange** objects are not sent.

-include_supercedures (isc)

Identifies the supercedure objects associated with the **BomChange** objects to send. If not specified, supercedure objects are not sent.

-include_pfmembers

Identifies the related part family members to be exported when handling part family templates.

-include_pftemplates

Identifies the related part family template to be exported when handling part family members.

-pf_bom_treatment

Identifies the part family objects associated with the assemblies to be exported. The argument must be used in conjunction with the **-include_bom** argument. Valid arguments are:

-members

Includes part family member components present in the assembly.

-templates

Includes part family template rather than part family member components.

-all

Includes both the part family member components and templates.

-none

Includes neither the part family member components nor the templates.

-qry-name

Runs a saved Query Builder that defines the list of objects to process. One or more **-qry_attr** and **-qry_val** pairs specify the attribute value pairs the objects must match to be processed.

-oaat

Forces the utility to process one object at a time. Normally, this utility processes all objects in a batch. If there is a failure on any top-level object, the entire batch fails. With this option, the other objects are successfully processed when one object fails. The log file indicates which object failed. The disadvantage to using this argument is the utility takes longer to process objects individually.

This argument is valid only if you specify the **send** function.

-continue_on_error (con)

Specifies processing is continue if there is an error on an optional object such as a reference or manifestation. This argument is not valid when transferring site ownership.

Outputs the error in a report file and continues processing the other items. The **-report** argument must be specified to see the error.

-batch_size (bs)

Specifies the number of objects per batch; a new process is created per batch. The default batch size is 1000. The value must be a positive integer. **-batch_size** is useful when processing thousands of objects, because it helps avoid memory and disk space shortage problems.

-report (rep)

Specifies to output a report to the specified file.

-user

Specifies the user ID.

-group

Specifies the group name.

-error_file (err)

Specifies the name of the output error report when sending assemblies.

-mapping_rule

Specifies the name of the *.xslt* file detailing the object transformation rules needed when transferring objects to a site with a different schema. You can create this *.xslt* file by running the **plmxml_tm_edit_xsl** utility with the **-action=export** argument on the transfer mode on which the transformation rules are attached.

You can specify mapping transfer modes for Advanced Multi-Schema Exchanger to use for Multi-Site TC XML importing and exporting using the **TC_tms_site_interop_transfer_mode** and **TC_tms_export_mapping_transfer_mode** preferences. These preferences do not affect Briefcase and PLM XML importing and exporting with Advanced Multi-Schema Exchanger.

-exclude_variant_options (evo)

Indicates all variant options are to be excluded during a send operation.

-batch_variant_options (bvo)

Indicates all variant options are sent separately in batch mode.

-batch_objects (bo)

Indicates all objects of the given classes are sent separately in batch mode. Separate each class name with a comma. The list cannot contain spaces. The following table is a list of supported classes for this argument:

Dataset	ImanRelation	PSOccurrence
Folder	MEAppearancePathNode	VariantExpression
Form	NamedVariantExpression	VariantExpressionBILock
	(Not supported by Multi-Site Collaboration.)	(Not supported by Multi-Site Collaboration.)

-batch_file (bof)

Indicates all objects of the classes in the specified file are sent separately in batch mode. List each class name separately on a line in the file. For a list of supported classes for this argument, see the table for the description of the **batch_objects** argument.

-include_dist_comp

Includes distributed components during import. This argument is valid only in conjunction with the **-remote_import** argument.

-lightweightobject (lwo)

Send the specified lightweight object unique identifier in the specified file. For example,

```
data_share -f=send -site=site2 -filename=uid.txt -
classoffile=Tagstring -lwo=ImanExportRecord
```

-log

Specifies detailed log information is written to the log file.

-checkpoint (cp)

Initiates a checkpoint transaction; that is, a transaction that can be restarted at the point of failing.

This argument 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 **stdout** to notify the user about unavailable sites, removes unavailable sites from the target site list, and then performs the operation for the available sites.

-compress_ind_files (cif)

Specifies compression mode to use to compress files in the export directory during a checkpoint transaction. If not specified, it creates a single large ZIP file. This argument is valid only with the **-checkpoint** argument. Valid values are:

- | | |
|----------|---|
| S | Creates a single large ZIP file. |
| I | Creates a ZIP file for each individual file, resulting in multiple ZIP files. |
| N | No files are compressed. |

-transaction_id (trid)

Specifies a 14-character transaction ID for a given checkpoint-related operation or for a fast sync transaction if used with the **-optionset** argument. A fast sync transaction provides improved performance when synchronizing data using the **data_sync** utility.

-tcx_data_load

Improves performance by disabling Multi-Site Collaboration business logic such as AM rule checks and in-process workflow data checks. This argument is intended only for migration scenarios.

Warning:

Use **-tcx_data_load** only in controlled migration scenarios. Disabling business logic in standard operation scenarios may cause unexpected changes in your data.

-tcx_data_load can be used with the **send**, **remote_import**, and **offline_export** functions.

-migration_sync

Reports unsynchronized data in support of migration scenarios. Used with **-tcx_data_load**, for example,

```
data_share -migration_sync -tcx_data_load -since=YYYY-MM-DD:HH:NN -
f=offline_export -dir=c:\outofsync
```

-status (stat)

Displays the status of a given transaction ID.

If the **-site** switch is given, only status of the given sites are displayed. If no **-site** switch is given, the status returned depends on whether the command is given at the site that initiated the checkpointed transaction or the site is the receiving end of the transaction. If the command is given at the initiating site and no **-site** switch is given, the status of the local site and all target sites is returned.

-cleanup_transaction (ct)

Removes transient data generated during a checkpoint transaction or fast synchronization transactions if used with the **-optionset** argument.

For checkpoint transactions, the transient data consists of the export data and supporting directories and files used to manage the transaction. You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have **delete** access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an error message is displayed to **stdout** and the utility returns a nonzero value.

For fast sync transactions, you must specify either the **-transaction_id** or **-before_last_process_date** argument.

-before_last_process_date (blpd)

Specifies the date used to determine which fast sync transactions to clean up. The date must be supplied in the following format:

```
YYYY-MM-DD:HH:NN:SS
```

YYYY represents the four-digit year value. MM represents the two-digit month value. DD represents the two-digit day of the month. HH represents a two-digit hour value from 0 to 23. NN represents a two-digit minute value from 0 to 59 and SS represents a two-digit second value from 0 to 59. The DD:HH:NN:SS are optional. If they are not specified, the utility uses 12:00 AM of the specified date.

Valid only with the **-cleanup_transaction** argument.

-restart (rs)

Restarts a given transaction at the point of failure.

Valid only with the **-f=send** function.

-commit_ixr (cmi)

Updates the export records at the owning site once the data is known to have been successfully imported at a target site.

Under normal conditions, the update of the export records are performed automatically by each subprocess that succeeds in completing the send operation to its assigned site. Use this function only if either of the following conditions occur:

- The failure occurs at the importing site, and the user performs the restart using the **item_import** utility.
- The failure occurs after the data is successfully imported by a target site, but a failure occurs just before or during the updating of the export records.

You must use at least one **-site** argument to identify the site or sites for which export records are to be updated.

You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have **read** access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an error message is displayed to **stdout** and the utility returns a nonzero value.

-list_transactions (lt)

Lists all uncleaned checkpoint transactions or fast sync transactions if used with the **-optionset** argument. An uncleaned transaction is one in which its transient data has not been deleted from the transfer area using the **cleanup_transaction** function.

- Active transactions can only be detected at the site that initiated it. The receiving end of a transaction is not able to determine if a transaction is active or not.
- The list of inactive transactions initiated by the local site and the list of transactions initiated by remote sites are based only on the contents of the transfer area of the node where this command is executed.
- The list of active transactions initiated by the local site is always complete because it is based on data stored in the local database.

You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have **read** access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an appropriate error message is displayed to **stdout** and the utility returns a nonzero value.

-all_subgroups

Exports all subgroups of the selected group. Parent groups are always exported. This argument is valid only when the **-class** or **-classoffile** argument is set to **Group**.

-all_roles

Exports all roles associated with the selected group. All roles for subgroups are included if the **-all_subgroups** argument is also specified. If not specified, only the default role is exported.

This argument is valid only when the **-class** or **-classoffile** argument is set to **Group**.

-all_groupmembers

When exporting a **User** object, exports all **GroupMember** objects related to the selected user. When exporting groups or groups and subgroups (**-all_subgroups** argument), exports all **GroupMember** objects related to any role that is exported.

You must specify one of the following arguments to use this argument:

```
-class=User
-classoffile=User
-class=Group
-classoffile=Group
```

If this argument is not specified when exporting **User** class objects, only the default group related to the user is exported.

If this argument is not specified when exporting **Group** class objects, no **GroupMember** objects are exported.

-optionset

Exports the data using the TC XML functionality. This method provides better performance for large data transfers and must be used when exporting 4GD data.

You can replicate 4GD objects to a remote site when you specify the **-optionset** argument with the **send** function. Optionally specify the transfer option set used for the export by setting **-optionset** to the set name. If you do not specify an option set, the utility uses **MultiSiteOptSet** as the default transfer option set value. Values of the options listed in the option set govern the object export. The option set must exist at the exporting site.

For 4GD objects, you can also specify the **-de_incl_rlz_bom** and **-workset_incl_relz_de** arguments.

-de_incl_rlz_bom

Exports the source objects of a design element (Type:**Cpd0DesignElement**). You must specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument. You must specify the **-optionset** argument and specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument.

-workset_incl_relz_de

Exports the source objects of a design element (**Cpd0DesignElement**) in a workset (**Cpd0Workset**). You must specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument. You must specify the **-optionset** argument and specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument.

-4gd_id

Specifies a 4GD object identifier or 4GD object pattern. The **-class=class-name** argument must be specified with the **-4gd_id** argument.

The utility maps the **-4gd_id** argument to the corresponding unique ID of the 4GD class, for example:

```
Class=Cpd0DesignElement, 4gd_id=cpd0design_element_id
```

A 4GD partition object and 4GD subset definition objects do not have a unique 4GD class ID. Therefore, using **-4gd_id** for partition objects or subset definition objects may result in transfer of multiple objects.

To export unique partition object use multifield key attributes supplied in the **-key** argument, see [Examples](#).

-parallelize

Specifies that parallel processing is enabled at the owning site and remote site when replicating data. **-parallelize** is valid only with the **-optionset** and **-send** functions. **-parallelize** is not valid with **-transfer**.

The number of additional to use processes at each site is defined by setting **-parallelize** to two comma-separated values. The first value specifies the number of additional processes to use at the owning site. The second value specifies the number of additional processes to use on the remote site. For example, to specify 5 additional processes on the owning site and 7 additional processes on the remote site, set **-parallelize** to **5,7**.

-parallelize supports replication to a single site. Ownership transfer is not supported.

Note:

Be judicious when specifying the number of additional processes to use at each site. If the workstation is powerful, with unused CPU cores and RAM, using parallel processes can improve performance. However, specifying parallel processing on less powerful workstations may overtax the workstations, draining system resources.

-override_options

Override the following default operations by setting the following values to **true**. (The default value of each is **false**.) Options and values are separated by colons. Multiple overrides are separated by commas.

opt_exclude_apn When exporting BOM structures, set to **true** to exclude APNs from being exported.

opt_include_rendering Set to **true** to include the rendering dataset in the exported data.

opt_include_ice When exporting **EngChange** lists, set to **true** to export **IncrementalChangeElement** items.

opt_exp_all_wso Set to **true** to export secondary objects for all relations. Use with care as this may result in a very large amount of data being exported.

-session_options

Override the following default session operations by setting the following values to **true**. (The default value of each is **false**.) Options and values are separated by colons. Multiple overrides are separated by commas.

forceUpdate Set to **true** to force an update during an import if a replica's last saved date (**lsd**)
OnImport value or last modified date (**lmd**) value is greater than the primary object's value.

opt_traverse Set to **true** to export only an island of data. Do not export across islands.
_by_island

-volume_file_list

Generates a text file listing the volume file paths associated with successfully imported or exported items. This information is particularly useful when migrating data. Used with **-offline_import** and **-meta_file**.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

- When sending objects to a specific user and/or group at a remote site using the **-owning_user** and **-owning_group** arguments, the following rules apply:
 - To use this utility, you must be a user with system administration privileges or be granted authorization by a user with system administration privileges.
 - If both the specified user and group exist at the importing site, the imported objects are owned by the user and group regardless of whether or not the user is a member of the group.
 - If only the user is specified or if the group is specified but does not exist at the importing site, the user's default group at the importing site is the owning group of the imported objects.
 - If only the group is specified or if the user is specified but does not exist at the importing site, the user context of the remote IDSM process is the owning user of the imported objects.
- If variant options are excluded using the **-exclude_variant_options** argument, it is implied that they cannot be sent separately in batch mode. Therefore, the **-exclude_variant_options** argument

cannot be used with either **-batch_variant_options**, **-batch_objects=variant-expression**, or with the **-batch_file** arguments when the given file includes the class name **VariantExpression**.

3. Any number of objects can be sent separately in batch mode. Class names of objects can be given in a comma-delimited list with the **-batch_objects** argument or listed in a file whose name is specified in the **-batch_file** argument.
4. The **-item_id**, **-name**, **-filename**, **-folder**, **-key**, **-itemKeyFile**, **-itemRevisionKeysFile**, and **-4gd_id** arguments are mutually exclusive
5. Traversal-free synchronization is not supported for remote import (**-f=remote_import**).
6. If the replica does not exist at the importing site, only the **-4gd_id** and **-item_id** arguments support the remote import function. (**-f=remote_import**).

For objects specified by **-filename**, multifeild **-key**, **-name**, or **-class** arguments, replicas must exist on the importing site.

All arguments supported for the send function (**-f=send**) with the **-optionset** argument are supported for remote import.

EXAMPLES

Required log-in information is omitted from the following examples.

- To send a list of items specified in a text file to two sites:

```
data_share -f=send -filename=my_item_list.txt -site=Site1 -site=Site2
```

- To send a list of items specified in a text file and output a report; continue processing if a nonfatal error is found:

```
data_share -f=send -filename=my_list.txt -site=Site1
-report=rep.txt -coe
```

- To send a precise assembly to a remote site:

```
data_share -f=send -item_id=xyz -rev=A -include_bom
-assert_precise
```

- To transfer ownership of a given item:

```
data_share -f=send -transfer -item_id=item123 -site=Site1
```

- To publish an assembly item and all its components using the latest revision rule to determine components:

```
data_share -f=publish -item_id=Engine100 -site=Ods1 -include_bom
```

- To publish an assembly item and all its components using the latest released revision rule to determine components:

```
data_share -f=publish -item_id=Item1 -site=Ods1 -include_bom
-latest_released
```

- To unpublish an assembly item and all its components from multiple ODS sites using the default revision rule **latest revision**:

```
data_share -f=unpublish -item_id=Item1 -site=Ods1 -site=Ods2
-include_bom
```

- To delete a publication record in the local database:

Note:

Use this only if the primary object has been deleted, but the publication record still exists.

```
data_share -f=delete_pubrec -item_id=ObsoleteItem1
```

- To list the authorized ODS sites:

```
data_share -f=list_ods
```

- To check availability of the authorized ODS sites:

```
data_share -f=check_ods
```

- To get publication information about a list of objects in a folder:

```
data_share -f=list_pub_info -folder=myFolder
```

- To send an item to a specific remote user and group:

```
data_share -f=send -item_id=xyz -site=Site1 -owning_user=joe
-owning_group=engg
```

- To send an item to a site using a different schema. In this scenario, Advanced Multi-Schema Exchanger is used to create the transformation rules to use when exchanging files. After attaching the rules to the transfer mode, the rules are extracted to a .xsl file using the **plmxml_tm_edit_xsl** utility with the **-action=export** argument on the transfer mode.

```
data_share -f=send -item_id=0001 site=site2
-mapping_rule=c:\mapping.xslt
```

- When publishing thousands of items and you get errors after publishing several hundreds or even thousands of items, reduce the batch size:

```
data_share -f=publish -item_id=A* -site=Site1 -batch_size=200
```

- To publish an engineering change object and all its associated change objects:

```
data_share -f=publish -item_id=CR0001 -site=Ods1 -include_bom
0-include_bc -include_supercedures
```

- To register an item ID with the central item ID registry:

```
data_share -f=register -item_id=myItem
```

- To find duplicate item IDs at another site:

```
data_share -f=find_duplicates -item_id=00* -site=Site1
```

- To list all objects that are checked out by remote users:

```
data_share -f=list_remote_co
```

- To list all objects that are checked out by user **justin** at **Site2**:

```
data_share -f=list_remote_co -user=justin -site=Site2
```

- To cancel check out of all objects by user **joseph** at **Site2**:

```
data_share -f=cancel_remote_co -user=joseph -site=Site2
```

- To list all replica objects that are checked out by local group **engg** from remote site **Site1**:

```
data_share -f=list_replica_co -site=Site1
```

- To cancel all replica objects that are checked out by user **davis** from **Site1**:

```
data_share -f=cancel_replica_co -user=davis -site=Site1
```

- To cancel remote checkout on a given item:

```
data_share -f=cancel_remote_co -item_id=item123
```

- To cancel remote checkouts on the datasets listed in the **dataset.lst** file:

```
data_share -f=cancel_remote_co -filename=dataset.lst -class=Dataset
```

- To cancel replica checkouts on the datasets in uniquely named folder:

```
data_share -f=cancel_replica_co -folder=unique_folder_xyz
```

- To exclude all variant options during a send operation:

```
data_share -f=send -item_id=CR0002 -site=remotel
-exclude_variant_options
```

- To batch send all variant options during a send operation:

```
data_share -f=send -item_id=CR0002 -site=remotel
-batch_variant_options -batch_size=10000
```

- To batch send one or more classes of objects during a send operation:

```
data_share -f=send -item_id=CR0002 -site=remotel
-batch_objects=class1,class2 -batch_size=10000
```

- To batch send one or more classes of objects given in a text file during a send operation:

```
data_share -f=send -item_id=CR0002 -site=remotel
-batch_file=my_list.txt -batch_size=10000
```

- To import an item from a remote site (**Site2**):

```
data_share -f=remote_import -site=Site2 -item_id=xyz
```

- To import the objects of an assembly using a file (**UIDs_list.txt**) containing a list of UIDs:

```
data_share -f=ri -filename=UIDs_list.txt -classoffile=Tagstring
```

- To initiate a checkpoint transaction at three specified sites:

```
data_share -f=send -checkpoint -item_id=item123
-site=Site2 -site=Site3 -site=Site4
```

- To export a role that does not have any group members from the **TopGrp1** group:

```
data_share -class=group -name=TopGrp1 -f=send
-all_groupmembers -site=Site2
```

- Include the rendering dataset and **IncrementalChangeElement** items in the exported data:

```
data_share -f=send -item_id=xxx -site=site2
-override_options=opt_incl_rendering:true,opt_include_ice:true
```

- Force updating of the data on the import site even though the last saved dates are the same for the primary and replica:

```
data_share -f=send -item_id=xxx -site=site2
           -session_options=ForceUpdateOnImport:true
```

- To return status for a transaction ID of **AhEZaOnRAAMfD** and no **-site** argument is specified:

```
data_share -f=status -trid=AhEZaOnRAAMfD
```

The output is similar to the following:

```
Site1: sending export data to all target sites (06-Nov-2007.14:31:28)
Site2: transmitting data (06-Nov-2007.14:35:31)
Site3: importing data (06-Nov-2007.14:34:29)
Site4: error 100107 - site not currently available (06-Nov-2007.14:32:26)
Site5: transaction complete (06-Nov-2007.14:40:10)
```

The time stamp represents the last time the status was updated.

- To return status at a receiving site and no **-site** argument is specified:

```
Site3: importing batch 5 out of 50 (06-Nov-2007.14:34:29)
```

At a receiving site, only the status of the local site is obtained; the status of other sites involved in a transaction are not available.

- To restart a given transaction for a given site:

```
data_share -f=send -transaction_id=AhEZaOnRAAMfD
           -restart -site=Site3
```

- To update export records at site **Site4** using a transaction ID of **AhEZaOnRAAMfD**:

```
data_share -f=commit_ixr -trid=AhEZaOnRAAMfD -site=Site4
```

- To clean up records with a transaction ID of **AhEZaOnRAAMfD**:

```
data_share -f=cleanup_transaction -trid=AhEZaOnRAAMfD
```

- To list checkpoint transactions:

```
data_share -f=list_transactions
```

The output is similar to the following:

Transactions initiated by local site:

AhEZaOnRAAAMfD - active

BxyzZaOnRAAXYZ - inactive

Transactions initiated by remote sites:

ZaOnRAAAYXCDA - Site3

- To send the **casCD001** collaborative design object to the **site2** site:

```
data_share -f=send -4gd_id=casCD001 -class=Cpd0CollaborativeDesign
-site=site2
```

- To transfer architecture breakdown structures between sites, you must execute the following steps sequentially:

1. Push the design assembly.

```
data_share -f=send -item_id=design-assembly-item_id -site=remote-site-id
-exclude=IMAN_reference -exclude=IMAN_based_on -exclude=IMAN_snapshot
-exclude=IMAN_3D_snap_shot -exclude=IMAN_external_object_link
-exclude=TC_Generic_Architecture -bo=VariantExpression, MEAppearancePathNode
-bs=1000 -ldv
```

2. Push the **LOUHOLDER** object and synchronize the BOM view revision.

```
data_share -f=send -item_id=<ITEM_ID> -site=<REMOTE_SITE> -include_bom
-exclude=IMAN_reference -exclude=IMAN_based_on -exclude=IMAN_snapshot
-bvo -bo=PSOccurrence -bs=20 -bvrsync -report=<IMAN_TMP_DIR>/rpt
```

3. Push the architecture breakdown structure excluding the NVEs.

```
data_share -f=send -item_id=<ITEM_ID> -rev=001 -site=<REMOTE_SITE>
-exclude=IMAN_3D_snap_shot -exclude=IMAN_reference
-exclude=IMAN_external_object_link
-exclude=IMAN_based_on -evo -bo=MEAppearancePathNode -bs=2000 -ldv
```

4. Push the NVEs.

```
data_share -f=send -item_id=<ITEM_ID> -rev=001 -site=<REMOTE_SITE>
-exclude=IMAN_3D_snap_shot -exclude=IMAN_external_object_link
-exclude=IMAN_based_on
-bo=MEAppearancePathNode -bs=2000 -ldv
```

- To send a 4GD object and output a report:

```
data_share -f=send -4gd_id=Ste2_DE0524 -cof=Cpd0DesignElement
-site=Site1 -optionset -report=4GD_rep.txt
```

- To send a list of 4GD objects specified in a text file using parallel processing and output a report:

```
data_share -f=send -filename=my_4GD_list.txt -cof=Cpd0DesignElement
-site=Site1 -optionset -parallelize=6,8 -report=rep.txt
```

- To send a specific 4GD partition object and output a report:

```
data_share -f=send class=Cpd0DesignElement
-key=ptn0partition_id=CD001_id,ptn0partition_scheme_type=SchemeFunctional_CD001,
partitionptn0source_object=partition_source_CD001,mdl0model_object=model_CD001,
mdl0revision_id=model__id=001/A site=Site1 -optionset -report=rep.txt
```

- To clean up fast synchronization transactions prior to specific last process date, list the available transactions to get the last process dates:

```
data_sync -optionset -lt
```

Clean up the transactions:

```
data_sync -optionset -ct -blpd=2012-12-18:20:10:00
```

- To use the **TC_cms_relation_optset_map** preference to include or exclude relations:
 1. Add the relation and option set to the **TC_cms_relation_optset_map** preference, for example:

```
IMAN_rendering, opt_rel_rendering
```

2. In the PLM XML/TC XML Export Import Administration application, expand the **TransferOptionSet**, click **MultiSiteOptSet**, and add the **opt_rel_rendering** option with the default value set to **false**.
3. Expand **ClosureRule**, click **MultiSiteDefaultCR**, and add the following clause:

```
CLASS:WorkspaceObject:CLASS:Dataset:RelationP2S:IMAN_rendering:
SKIP:$opt_rel_rendering==false;
```

This clause states, from any **WorkspaceObject**, find the dataset using **IMAN_rendering** relation, and when the relation is **opt_rel_rendering**, skip the dataset during export. It means the default is to always exclude the **IMAN_rendering** relation for an exported object.

4. To include the **IMAN_rendering** in the export using the **data_share** utility, type:

```
data_share -include=IMAN_rendering -optionset
```

dataset_cleanup

Repairs corrupted datasets and removes orphaned datasets and revision anchors.

Caution:

Siemens Digital Industries Software recommends that you run this utility only when there is no other activity on the database.

PROBLEM IDENTIFIERS

A dataset is identified as corrupted if any of the following problems are found:

- Dataset has no reference to an **ImanFile** object.
- Dataset has reference to an **ImanFile** object, but the corresponding operating system file does not exist and the dataset is not archived.
- Dataset is an orphan (that is, the dataset refers to the anchor but the anchor does not go to dataset).
- Anchor refers to datasets that do not exist.
- Anchor size = 0.

OBJECT CLEANUP RULES

A dataset object is reattached to revision anchor if it is an orphan but is referenced by some other objects, or deleted if it meets the following criteria:

- Dataset is an orphan and is not referenced.
- Dataset is not archived and the associated operating system file does not exist.

ANCHOR CLEANUP RULES

The **dataset_cleanup** utility repairs dataset revision anchors as follows:

- If the anchor refers to nonexistent datasets, the references are removed from the anchor.
- If the anchor size = 0, the anchor is deleted.

SYNTAX

```
dataset_cleanup [-u=user-id {-p=password | -pf=password-file} -g=group]
[-reportCorruption | -fixCorruption | -reportBadAnchor | -fixBadAnchor | -reportRestorable |
-fixRestorable | -reportOrphan | -deleteOrphan] -h
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-reportCorruption

Generates a report of all corrupted dataset versions and anchors.

The report file, *dataset_cleanup_actions.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-fixCorruption

Either fixes or removes all corrupted dataset versions and anchors.

A report file of corrections, *dataset_cleanup.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-reportBadAnchor

Generates a report of all bad anchors.

The report file, *dataset_cleanup.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-fixBadAnchor

Fixes all bad dataset anchors.

The report file, *dataset_cleanup_fix_anchor.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-reportRestorable

Generates a report of all restorable orphaned datasets.

The report file, *dataset_cleanup.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-fixRestorable

Fixes all restorable orphaned datasets.

The report file, *dataset_cleanup_restore.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-reportOrphan

Generates a report of all invalid and orphaned datasets.

The report file, *dataset_cleanup.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-deleteOrphan

Deletes all invalid and orphaned datasets.

The report file, *dataset_cleanup_delete.log*, is generated in the running directory and overwrites any existing *dataset_cleanup.log* file.

-h

Displays help for this utility.

LEGACY

```
dataset_cleanup [-u=user-id {-p=password | -pf=password-file} -g=group]
-rf=file-name | -if=file-name
[-of=log-file-name] [-b=beginning-anchor]
[-e=ending-anchor] [-start_date=start-date] [-end_date=end-date] -h
```

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-rf

Creates a report file listing the corrupted datasets. If the location of the output file should be other than the current folder, include the path as well as the output file name.

-if

Uses the report file as input to purge corrupted and orphaned datasets and repair revision anchors.

Caution:

This option deletes orphaned datasets. Carefully examine the input file and remove from the list any orphaned datasets that you wish to retain. It is also an administrative best practice to paste important datasets into a folder to prevent inadvertent deletion.

-of

Cleans up and logs the results to a log file. This argument must be supplied if the **-if** argument is used but is optional with the **-rf** argument.

-a

Specifies that corrupt anchors (those that are orphaned and are not referenced by a dataset) be deleted and a message be provided.

-b

Specifies the first revision anchor of a contiguous series to be repaired. The default value is **1**.

-e

Specifies the last revision anchor of a contiguous series to be repaired. The default value is **last**.

A revision anchor is an object that keeps track of a set of revisions of some object. One such class of objects is datasets.

-start_date

Specifies the starting date to search for datasets that have been modified from this date. Use this argument with the **-end_date** argument.

The format of the date is “*DD-MMM-YYYY HH:MM:SS*” and must be inside the double quotes because of the space between the year and the hour. This argument is used only with the **-rf** argument.

-end_date

Specifies the ending date to search for datasets that have been modified until this date. This argument is optional and is used only with the **-start_date** argument. If this argument is not specified, the end date is the current date.

The format of the date is “*DD-MMM-YYYY HH:MM:SS*” and must be inside the double quotes because of the space between the year and the hour. This argument is used only with the **-rf** argument.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment..

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

None.

EXAMPLES

- To generate a report file called **myreportfile** listing corrupted dataset objects, enter the following command on a single line:

```
$TC_ROOT/bin/dataset_cleanup -u=admin-user -p=admin-password -g=dba
-rf=myreportfile
```

- To run the **dataset_cleanup** utility using the **myreportfile** file as input, enter the following command on a single line:

```
$TC_ROOT/bin/dataset_cleanup -u=admin-user -p=admin-password -g=dba
-if=myreportfile -of=mylogfile
```

- On a database with 1000 dataset revision anchors, you could run the **dataset_cleanup** utility as follows:

```
$TC_BIN/dataset_cleanup -u=admin-user -p=admin-password -g=dba -b=1
-e=500
-rf=dataset_cleanup_500.report
$TC_BIN/dataset_cleanup -u=admin-user -p=admin-password -g=dba
-b=501 -e=1000 -rf=dataset_cleanup_1000.report
```

- To purge all datasets with modification dates between Oct-01-2020 and Oct-10-2020:

```
dataset_cleanup -u=admin-user -p=admin-password -g=dba
-start_date="01-OCT-2020 00:00:00" -end_date="10-Oct-2020 00:00:00"
-rf=ttt.txt
```

- To purge all datasets with modification dates from Oct-01-2020 to the current date:

```
dataset_cleanup -u=admin-user -p=admin-password -g=dba
-start_date="01-OCT-2020 00:00:00" -rf=ttt.txt
```

CLEANING UP DATASETS AND REPAIRING REVISION ANCHORS

Perform the following steps to clean up corrupted datasets:

1. Use the **dataset_cleanup** utility to generate a report file called **myreportfile** listing the corrupted dataset objects in the database by entering the following command on a single line:

```
$TC_ROOT/bin/dataset_cleanup -u=admin-user -p=admin-password -g=dba
-rf=myreportfile
```

The report file contains a list of corrupted datasets sorted by **Object_UID**. The report also contains the problem identifier, dataset name, and ownership.

If the **-a** argument is specified on the command line, the utility deletes the corrupt anchors and displays a message to the user. If the **-a** argument is not supplied, a message is displayed indicating that the anchor was skipped and the **-a** option should be used.

You must review the report file and decide which datasets, if any, should not be purged from the database.

2. Use a text editor to remove any references to dataset objects that should not be purged from the database from the report file.

3. Run the **dataset_cleanup** utility using the **myreportfile** file as input to purge corrupted dataset objects from the database or fix anchors and log the results to the **mylogfile** file by entering the following command on a single line:

```
$TC_ROOT/bin/dataset_cleanup -u=admin-user -p=password -g=dba  
-if=myreportfile -of=mylogfile
```

The utility attempts to fix the revision anchor, attach the dataset to another revision anchor, or purge the datasets from the database.

A final output report is generated showing the results for each dataset. The report displays the following message if the operation is successful:

```
problem deleted
```

If the operation is unsuccessful, the following message is displayed:

```
could not delete error stack number
```

4. Display information about the dataset cleanup process by entering the following command:

```
ps -ef | grep data
```

5. Kill the dataset cleanup process by entering the following command:

```
kill -9 PID
```

PID is the operating system process ID returned in step 4.

data_sync

Synchronizes copies of objects at remote sites with the latest version of the primary object. It also updates publication records when republishing objects. In **verify** mode, the utility checks the existence of exported objects at the remote sites; if a copy no longer exists at the remote site, the corresponding import export record is deleted from the owning site.

The behavior of this utility is controlled by the **TC_force_remote_sites_exclude_files** preference. If this preference is set to **true**, the replica files stored in the remote site file server cache (FSC), otherwise the replica files are store in the remote system volume.

The **data_sync** utility uses import export records (IXRs) and publication audit records (PARs), which are attached to the primary copy of an object, to determine whether or not to synchronize a copy or the publication record in the ODS. These records contain information on when the object was last sent to a particular site or last published to an ODS. It then compares these dates with the object's last-modified date and decides whether or not to synchronize the object. Thus, only those objects that were modified since the last successful run of the utility are updated.

When updating multiple sites and not all sites are operational, the **data_sync** utility updates the sites that are available but remembers, using the IXRs and the PARs, which ones were unavailable so they can be updated next time.

Once the utility determines which objects and sites to synchronize, it uses the basic Multi-Site Collaboration mechanisms (export, import, IDSM, and ODS) to accomplish its task. For this reason, Siemens Digital Industries Software recommends that the **data_sync** utility be run in batch mode during off hours so that it does not compete for computing and network resources during business hours.

The utility also supports TC XML transfers for 4th Generation Design (4GD) data. The 4GD relation data mapping is controlled by the **TC_cms_relation_optset_map** preference. You use this preference when you want to control the relations that are included or excluded when a 4GD object is replicated.

The **data_sync** utility supports part family templates and members. It also supports organization classes, specifically, **Role**, **User**, **Group**, and **Person** classes.

Siemens Digital Industries Software recommends the following practices when using the **data_sync** utility. The term *one at a time* means one command line invocation. This implies that your script for running **data_sync** consists of several lines invoking the **data_sync** utility.

- Synchronize one site at a time and use the default revision selector of **-same_as_last_export**. This allows you to use the Smart Sync capability which synchronizes only the revisions and attachments that the remote user specified when replicating an item.
- Synchronize one class at a time starting with the largest unit, which is **Item**, and down to the smallest units such as **Dataset** and **Form**.
- Always use the **-since** switch with the **-class** switch. This results in improved memory efficiency because replicated objects that have not been modified for some time are excluded from the initial

search for objects to be synchronized. Ideally, the date given to the **-since** switch should be the exact date and time of the last successful run of **data_sync**. However, if you are not sure about the date and time, use a date and time that you know is prior to the last successful run.

- When dealing with thousands of objects, **data_sync** tends to slow down as it loads more and more objects in memory. It handles this problem by cascading its work over several sub-processes. When the original process reaches its batch size, it starts another process and then terminates itself; the sub-process continues where its parent process left off. When it reaches its own batch size, it creates another sub-process, and so on. The optimum batch size varies for each installation depending primarily on the memory (both main memory and virtual memory), so you must determine the optimum batch size for your installation.

One tool that can help you do this is the use of the **-log** switch that records all significant events in the **data_sync** log file, the file with the **.log** extension. By analyzing the log file, you can detect at what point **data_sync** begins to slow down so you can then adjust the batch size accordingly. Note that the use of the **-log** switch itself can affect the overall efficiency of **data_sync** so you should turn off the switch once you have determined your optimum batch size.

- The synchronization process can put a heavy load on the network and the systems so **data_sync** should be scheduled during non-busy hours such as nights and weekends. Typically, you should run the synchronization script as a cron job to be started at night.
- It is not necessary to have a separate verify run before synchronization because **data_sync** always performs a verification before synchronization. View a separate verification run as a cleanup procedure and run it only when the network and the systems are not busy, such as on weekends.
- Do not use the **-disable_modified_only** switch unless there is a known problem with the default synchronize modified objects only mode.
- If you typically share whole assemblies with a site, it is best to use the **-filename** switch to synchronize specific assemblies and use the **-include_bom** switch to synchronize any modified components. Note that you may have to use the **-force** switch in the event the item itself was not modified but you want to synchronize modified components.

The behavior of the utility for project relationships on replica objects can be controlled by the **TC_sync_projects_with_owning_site** preference. This preference is not created by the Teamcenter installation process. To change the default behavior of shared project relationships, you must create the preference.

AVOID OUT-OF-MEMORY ERRORS

To avoid possible out-of-memory errors when you are replicating architecture revisions, Siemens Digital Industries Software recommends you exclude **MEApperancePathNode** (APN) objects by setting the following environment variable:

```
TC_EXCLUDE_APN=TRUE
```

This excludes APN objects from the synchronization. It may also exclude associated object, such as JT for promoted and deformed bodies. You can synchronize the APN objects by using the **sync_product_apns** utility.

If you do not set this environment variable, or you set its value to **FALSE**, you can use the batch feature of this utility to overcome memory limitations when synchronizing large numbers of APNs. For example, you may specify:

```
-batch_objects=MEAppearancePathNode -batch_size=5000
```

When setting the batch size, consider the number of APNs generated by your business processes and the amount of memory available on your system.

SYNTAX

```
data_sync [-u=user-name {-p=password | -pf=password-file} -g=group]
{-class=class-name [-filename=file-name | -itemKeyFile=file-name] |
-item_id=template | -key=keyAttr1=keyVal1
[,keyAttr2=keyVal2...,keyAttrN=keyValN]} [-OnlyVIS]
{-site=site-name -sync | -republish | -verify}
[-f=sync | republish | verify] [-status] [-commit_ixr]
[-cleanup_transaction [-transaction_id=transaction-id |
-before_last_process_date=date ] | list_transactions] [-pull] [-update]
[-replacement_site=site-replacing-extinct-site] [-stubs_only] [-sync_file_stubs]
[-force]
[-report[=file-name] [-format[=CSV] [-separator[="|" | ","]]]]
[-exclude_files] [-disable_modified_only]
[-exclude=relation-type1 -exclude=relation-type2...]
[-exclude_security_update]
[-include=relation-type1 -include=relation-type2...]
[-include_bom] [-classoffile=class-name] [-revision-selector]
[-latest_ds_version] [-assert_extinct_ods] [-assert_extinct_site]
[-exclude_folder_contents] [-since=YYYY-MM-DD:HH:NN]
[-qry_name=query-name -qry_attr=attr-name1 -qry_val=attr-value1[-qry_attr=attr-name2
-qry_val=attr-value2...]]
[-batch_size=number-of-objects-per-batch]
[-deferred_batch_size=batch-size-for-deferred-objects]
[-batch_objects=list-of-deferred-classes]
[-batch_file=file-name-listing-deferred-classes]
[-nonbulk]
[-verbose] [-log]
[-checkpoint [-compress_ind_files={S | I | N} ] ]
[-transaction_id] [-restart]
[-optionset | -optionset=optionset-name [-de_incl_rlz_bom]
[-workset_include_relz_de] [-4gd_id=object-id -class=4gd-class-name] ]
[-include_4gd_baseline_content]
[-override_options=option1:value1,option2:value2, ...]
[-session_options=option1:value1,option2:value2, ...]
```

[-bp]
[-h]

ARGUMENTS

Entries in parentheses are accepted abbreviations for arguments.

-u

Specifies the user ID. This is generally a user with administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

Caution:

For HTTP enabled sites, remote site operations log on using the default group for the user supplied with the **-u** argument. Any value supplied with the **-g** argument is ignored.

-pull

Specifies synchronization starts in **pull** mode, that is, from a replica site.

For 4GD objects, you must specify the **-optionset** argument. This causes the utility to pull the data as a TC XML payload.

-class

Specifies the class of objects to be searched to determine what objects need synchronization. This does not mean that all objects of the given class are synchronized; only those that were modified since the last time they were exported to the given site(s) are synchronized. See restrictions **1** and **2**.

-filename (fn)

Specifies the name of the input file containing IDs or names of objects to update. See restriction 1.

-itemKeyFile

Specifies the name of the input file containing the list of item key strings of the items you want to update. The following listing shows sample content of a file for updating a list of items:

```
item_id=export_001
item_id=M2Item1_001,object_name=M2Item_name1,object_type=M2Item1
```

If the item key file has 4GD object key strings, the corresponding 4GD class must be supplied using the **-classoffile(cof)=** argument. The following listing shows sample content of a file for updating 4GD data:

```
4gd_id=DE_Export_001
4gd_id=DE00001_ID,object_name=DE00001_Name,object_desc=DE00001_Desc
```

The **4g_id** entry maps to the corresponding unique ID of the 4GD class, for example:

```
Class=Cpd0DesignElement, 4gd_id=cpd0_design_element_id
```

-item_id (item)

Specifies the ID or template of items to update. See restriction 1.

-key

Specifies the item keys of the items to update, the template of the item keys, or the 4GD object key. It is mutually exclusive with the **-item_id** argument. Use the following format:

```
[keyAttr1=keyVal1] [,keyAttr2=keyVal2]...[,keyAttrN=keyValN]
```

To find the key of an object, use the **get_key_string** utility.

-OnlyVIS

Specifies synchronization of only visualization datasets attached directly or indirectly to a replicated item revision with status. See restriction 8.

If you use the **-OnlyVIS** argument, the **-batch_size** value defaults to **500**.

-f

Specifies the function to be performed; define one of the following functions:

sync

Initiates the update process. See restrictions 2 and 4.

republish (repub)

Republishes objects that have been modified since last published. See restriction 2.

verify (veri)**Note:**

Siemens Digital Industries Software recommends that you always use the **-item_id=*** argument with the **verify** function. If you use the **-class=item** argument with the **verify** function, it processes only the items that have been modified after their last export.

When used with the **-update** argument, deletes the IXRs of objects for which replicas do not exist at the remote sites.

When used without the **-update** argument, generates a report. See restriction 2.

The report returns the following verification verdict codes:

0	Object does not exist.
1	Object exists as a primary copy.
2	Object exists as a replica.
3	Object was replaced by a POM stub.
status (stat)	Displays the status of a given transaction ID.

If the **-site** argument is given, only status of the given sites is displayed. If no **-site** argument is given, the status returned depends on whether the command is given at the site that initiated the checkpointed transaction or the site is the receiving end of the transaction. If the command is given at the initiating site and no **-site** argument is given, the status of the local site and all target sites is returned.

commit_ixr (cmi)

Updates the export records at the owning site once the data is known to have been successfully imported at a target site.

Under normal conditions, the update of the export records are performed automatically by each subprocess that succeeds in completing the send operation to its assigned site. Use this function only if either of the following conditions occur:

- The failure occurs at the importing site, and the user performs the restart using the **item_import** utility.
- The failure occurs after the data is successfully imported by a target site, but a failure occurs just before or during the updating of the export records.

You must use at least one **-site** argument to identify the site or sites for which export records are to be updated.

You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have **read** access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an error message is displayed to **stdout** and the utility returns a nonzero value.

cleanup_transaction (ct) Removes transient data generated during a checkpoint transaction or fast sync transactions if used with the **-optionset** argument.

For checkpoint transactions, this transient data consists of the export data and supporting directories and files used to manage the transaction. You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have **delete** access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an error message is displayed to **stdout** and the utility returns a nonzero value.

For fast sync transactions, you must specify either the **-transaction_id** or **-before_last_process_date** argument.

-before_last_process_date (blpd) Specifies the date used to determine which fast sync transactions to clean up. The date must be supplied in the following format:

YYYY-MM-DD:HH:NN:SS

YYYY represents the four-digit year value. *MM* represents the two-digit month value. *DD* represents the two-digit day of the month. *HH* represents a two-digit hour value from 0 to 23. *NN* represents a two-digit minute value from 0 to 59 and *SS* represents a two-digit second value from 0 to 59. The *DD:HH:NN:SS* are optional. If they are not specified, the utility uses 12:00 AM of the specified date.

Valid only with the **-cleanup_transaction** argument.

list_transactions (lt) Lists all uncleaned checkpoint transactions. An uncleaned transaction is one in which its transient data has not been deleted from the transfer area using the **cleanup_transaction** function.

- Active transactions can only be detected at the site that initiated it. The receiving end of a transaction is not able to tell if a transaction is active or not.
- The list of inactive transactions initiated by the local site and the list of transactions initiated by remote sites are based only on the contents of the transfer area of the node where this command is executed.
- The list of active transactions initiated by the local site is always complete because it is based on data stored in the local database.

You must execute this function at a node or host that has direct access to the transfer area where the export was performed (if at initiating site) or where the data was transmitted to by the owning site (if at receiving site). You must also have *read* access to the operating system directory where the export data is placed within the transfer area. If these conditions are not met, an appropriate error message is displayed to **stdout** and the utility returns a nonzero value.

-assert_extinct_ods (aao)

Deletes all publication audit record (PAR) objects from the local database for an ODS that no longer exists. This removes any record about objects previously published to the ODS and makes it possible to delete the published objects at a later time. Only the **-site** and **-login** switches are required. This is valid only with the **-f=verify** argument. See restriction 5.

-assert_extinct_site (aes)

Deletes all import export record (IXR) objects from the local database for a site that no longer exists. This removes any record of objects previously exported to the site and makes it possible to delete the exported objects at a later time. Only the **-site** and **-login** arguments are required. This is valid only with the **-f=verify** argument. See restriction 5.

-replacement_site (rs)

Specifies the name of the site that replaces the site to be extinct. This argument is valid only with the **-assert_extinct_site** argument. All objects owned by the extinct site are redirected to the replacement site.

-stubs_only

Specifies that stubs are processed only when the **-replacement_site** argument is specified. This argument is valid only with the **-assert_extinct_site** argument.

-sync_file_stubs

Processes dataset files excluded from export and updates remote stubs if needed. This argument is valid only when the **-class=ImanFile** argument is specified.

-update (upd)

Performs a database update. Must be given in order for the **-f=sync**, **-f=republish**, or **-f=verify** to occur; otherwise, it only does a dry run and generates a report. See restriction 4.

-report

Generates a synchronization report. If a file name is not supplied, the report is displayed in a shell. When published to a file, the default format is as a comma-delimited .csv file. To use a pipe (|) as a separator, include **-format** and **-separator = "|"**. For example,

```
data_sync -u=tcadmin -p=password -item_id=iteamA012 -sync -optionset
-update -site=site2 -report=report.csv -format=CSV -separator="|"
```

-site

Specifies the Teamcenter site to update. This argument can be used multiple times in the command line to synchronize with multiple name-identified sites.

-exclude_files (exf)

Excludes dataset files. See restriction 6.

-exclude

Excludes the specified relation type. This argument may be given multiple times and must use the database name (not the display name) of the relation type. See restriction 6.6.

For best TC XML performance, use a closure rule for relation instead of include/exclude.

-exclude_security_update

Excludes items with only changed project or license security data. When these items are not required to be synchronized, excluding them can significantly improve synchronization performance. Items with other changes in addition to project or license security data changes will be synchronized.

-include

Includes the specified relation type. This argument may be given multiple times and must use the database name (not the display name) of the relation type. Use this argument to force the inclusion of a relation type that may have been excluded during the last export.

For best TC XML performance, use a closure rule for relation instead of include/exclude.

-exclude_folder_contents (efc)

Excludes the contents of a folder. Intended for use with NX part families where family members are stored in a folder that is related to the item.

-include_bom (bom)

Synchronizes all components of an assembly. This synchronization includes any newly added components to the existing assembly. See restriction 6.

-disable_modified_only (dmo)

Disables the default behavior of synchronizing subobjects inside an item only if they were modified since the last time the item was exported.

Normally, this argument is not used. See restriction 6.

-revision-selector

Valid only if both the **-f=sync** and **-update** arguments are specified. Choose one of the following revision selectors:

-all_revisions

Synchronizes all revisions.

-latest_revision

Synchronizes only the latest revision, regardless of the release status. This is the default if no revision selector is specified and more than one site is to be synchronized. If synchronizing only one site, the default selector is **same_as_last_export**.

-latest_working

Synchronizes only the latest working (unreleased) revision.

-latest_released

Synchronizes only the latest released revision with any release status.

-latest_working_or_any

Synchronizes the latest working revision; if no working revision, synchronizes the latest released revision of any release status.

-release_status = release-status-type

Synchronizes only the latest released revision with the specified release status type.

-all_released_revs

Synchronizes all revisions with a release status including in-process item revisions.

-same_as_last_export

Synchronizes using the options used the last time the item was exported. This is the default if no revision selector is specified and only one site is being synchronized. If synchronizing multiple sites, the default selector is **latest_revision**.

-include_pfmembers

Identifies the related part family members to be exported when handling part family templates.

-include_pftemplates

Identifies the related part family template to be exported when handling part family members.

-pf_bom_treatment

Identifies the part family objects associated with the assemblies to be exported. The argument must be used in conjunction with the **-include_bom** argument. Valid arguments are:

-members

Includes part family member components present in the assembly.

-templates

Includes part family template rather than part family member components.

-all

Includes both the part family member components and templates.

-none

Includes neither the part family member components nor the templates.

-latest_ds_version (ldv)

Synchronizes only the latest version of datasets. See restriction 6.

-force

Synchronizes objects regardless of whether they were modified since the last time they were exported. See restriction 3.

-since

Synchronizes only those objects modified since the specified date and time, which must be specified in *YYYY-MM-DD:HH:NN* format, where *YYYY* is the year; *MM* is the month number from 1 to 12; *DD* is the day from 1 to 31; *HH* is the hour from 0 to 23, and *NN* is the minute from 0 to 59. *HH* and *NN* are optional and default to zero, which indicates 12 a.m. of the given date. This is valid only with the **-class** argument.

-verbose

Displays maximum amount of information when the utility is run in verbose mode. Typically, nonverbose utility sessions only display error messages. Do not abbreviate this argument to **-v**.

-log

Places detailed information in the **data_sync.log** file. The information includes the start and ending time for each step performed by the **data_sync** utility. Use this argument to analyze the performance of the utility.

-checkpoint (cp)

Initiates a checkpoint transaction, that is, a transaction that can be restarted at the point of failing. This argument is valid only when both **-f=sync** and **-update** are specified. If specified without the **-update** argument, this argument is ignored.

Valid only with **-f=sync**.

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 **stdout** to notify the user about unavailable sites, removes unavailable sites from the target site list, and then performs the operation for the available sites.

compress_ind_files (cif)

Specifies compression mode to use to compress files in the export directory during a checkpoint transaction. If not specified, creates a single large ZIP file. This argument is valid only with the **-checkpoint** argument. Valid values are:

- **S**

Creates a single large ZIP file.

- **I**

Creates a ZIP file for each individual file, resulting in multiple ZIP files.

- **N**

No files are compressed.

-transaction_id (trid)

Specifies a 14-character transaction ID for a given checkpoint-related operation or fast sync transaction.

-optionset

Sends the data using the TC XML functionality. This method provides better performance for large data transfers and must be used when exporting 4GD data.

You can replicate 4GD objects to a remote site when you specify the **-optionset** argument with the **send** function. Optionally specify the transfer option set used for the export by setting **-optionset** to the set name. If you do not specify an option set, the utility uses **MultiSiteOptSet** as the default transfer option set value. Values of the options listed in the option set govern the object export. The option set must exist at the exporting site.

For 4GD objects, you can also specify the **-de_incl_rlz_bom** and **-workset_incl_relz_de** arguments.

-de_incl_rlz_bom

Sends the source objects of a design element (Type:**Cpd0DesignElement**). You must specify the **-optionset** argument and specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument.

-workset_incl_relz_de

Sends the source objects of a design element (**Cpd0DesignElement**) in a Workset (**Cpd0Workset**). You must specify the **-optionset** argument and specify a 4GD object using the **-4gd_id**, **-key**, or **-itemKeyFile** argument.

-4gd_id

Specifies a 4GD object identifier or 4GD object pattern. The **-class=class-name** argument must be specified with the **-4gd_id** argument.

The utility maps the **-4gd_id** argument to the corresponding unique ID of the 4GD class, for example:

```
Class=Cpd0DesignElement, 4gd_id=cpd0design_element_id
```

A 4GD partition object and 4GD subset definition objects do not have a unique 4GD class ID. Therefore, using **-4gd_id** for partition objects or subset definition objects may result in the update of multiple objects

To export unique partition object use multifield key attributes supplied in the **-key** argument, see [Examples](#).

-restart (rs)

Restarts a given transaction at the point of failure.

Valid only with the **-f=send** function.

-qry-name

Runs a saved Query Builder that defines the list of objects to process. One or more **-qry_attr** and **-qry_val** pairs specify the attribute value pairs the objects must match to be processed.

-batch_objects (bo)

Specifies a list of comma-separated deferred classes. The list must not contain spaces.

-batch_file (bof)

Specifies the file name of a text file containing a list of deferred classes. Each class name is contained on a separate line.

-batch_size (bs)

Specifies the number of objects to synchronize per batch. A new process is created for each batch. All workspace objects (not just items) that are synchronized are considered part of a batch. The default batch size is 2000. The maximum value you can specify is 99999. If you enter a value greater than 99999, the utility sets the value of **-batch_size** to the default.

To process items one item at a time, use the **-non-bulk** argument.

-deferred_batch_size (dbs)

Specifies the number of objects per batch; a new process is created per batch. The default value is 2000. This value must be a positive integer. Use this argument to process thousands of objects to avoid memory and disk shortage problems.

The following classes are supported for deferred objects:

- **Dataset**
- **Folder**
- **Form**
- **ImanRelation**

- **MEAppearancePathNode**
- **NamedVariantExpression**
- **PSOccurrence**
- **VariantExpression**
- **VariantExpressionBlock**

-nonbulk

Process one object at a time when synchronizing a list of objects. By default, **data_sync** runs with **-batch_size** set to a value of **2000**. Overriding this default operation with **-nonbulk** may result in slower performance, but can aid debugging when researching batch synchronization issues.

-override_options

Override the following default operations by setting the following values to **true**. (The default value of each is **false**.) Options and values are separated by colons. Multiple overrides are separated by commas.

opt_exclude_apn When syncing BOM structures, set to **true** to exclude APNs from being synchronized.

opt_include_rendering Set to **true** to include the rendering dataset in the synced data.

opt_include_ice When syncing **EngChange** lists, set to **true** to sync **IncrementalChangeElement** items.

opt_exp_all_wso Set to **true** to sync secondary objects for all relations. Use with care as this may result in a very large amount of data being synchronized.

-session_options

Override the following default session operations by setting the following values to **true**. (The default value of each is **false**.) Options and values are separated by colons. Multiple overrides are separated by commas.

forceUpdateOnImport Set to **true** to force an update during a sync if a replica's last saved date (**Isd**) value or last modified date (**Imd**) value is greater than the primary object's value.

opt_traverse_by_island Set to **true** to sync only an island of data. Do not sync across islands.

-bp

Displays best practices information.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in *Teamcenter Utilities*.

RESTRICTIONS

1. One of the following arguments must be supplied: **-class**, **-filename**, or **-item_id**.
2. One of the following arguments must be supplied: **-f=sync**, **-republish**, or **-f=verify**.
3. The **-force** argument can only be used along with the **-filename** or the **-item_id** argument. It does not function when used in combination with the **-f=verify** argument.
4. Unless the **-update** argument is given, the **data_sync** utility generates only reports.
5. The **-assert_extinct_site** and **-assert_extinct_ods** options can only be used with the **-f=verify** argument.
6. The **-exclude_files**, **-exclude=**, **-include_bom**, **-disable_modified_only** and **-latest_ds_version** options can be used if both the **-f=sync** and **-update** arguments are supplied.
7. The **-classoffile** argument currently supports only the **Item**, **ItemRevision**, **Dataset**, **Form**, **Folder**, **Role**, **User**, **Group**, and **Person** classes.
8. The **-include** and **-update** arguments must be supplied with the **-OnlyVIS** switch.
9. To use this utility, you must be a user with system administration privileges or be granted authorization by a user with system administration privileges.
10. For 4GD data, traversal-free synchronization is not supported in pull mode.

All arguments supported for push mode with the **-optionset** argument are supported in pull mode.

EXAMPLES

Required log-in information is omitted from the following examples.

- To generate a report of items that must be synchronized for a given site:

```
data_sync -class=Item -site=Site1 -f=sync
```

The report is output to **stdout**. No synchronization is performed.

- To synchronize all items copied to a site and output a report to a file:

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

(The default revision selector, **-same_as_last_export**, is used.)

- To synchronize the latest released revisions of items:

```
data_sync -class=Item -site=Site1 -f=sync -update -latest_released
```

- To synchronize all forms and datasets:

```
data_sync -class=Form -class=Dataset -site=Site1 -f=sync -update
```

- To republish all previously published items to the **Mfg_ODS** ODS:

```
data_sync -class=Item -site=Mfg_ODS, -f=republish -update
```

- To check if datasets copied to the **Design_Center** site still exist and delete the IXR from the primary if a copy is no longer there:

```
data_sync -class=Dataset -site=Design_Center -f=verify -update
```

- To force synchronization of a list of items specified in a text file copied to a site and output the report to a file:

```
data_sync -filename="/myhome/itemlist.txt" -classoffile=Item  
-site=Site1 -f=sync -update -force -report=report.lst
```

- To force synchronization of a single item or items that match a template:

```
data_sync -item_id=Eng* -site=Site1 -f=sync -update -force  
-report=rep.lst
```

- To synchronize an out-of-date 4GD design element and output the report to a file:

```
data_sync -4gd_id=Ste2_DE0486 -class=Cpd0DesignElement  
-site=Site1 -optionset -sync -update  
-report=4GD_update_rep.txt
```

- To destroy all the export records to a known extinct site:

```
data_sync -u=Tc-admin-user -p=password -g=group -site=XSite -f=verify  
-update -assert_extinct_site
```

- To destroy all the publication records to a known extinct ODS site:

```
data_sync -u=Tc-admin-user -p=password -g=group -site=XSite -f=verify
-assert_extinct_ods
```

- To destroy export records, BVRs, and attachments of specific deleted replica item revisions:

```
data_sync -site=S1 -f=verify -update -fn=mylist
-cof=ItemRevision
```

The **mylist** file has item revision names in the following format: **item123/A**

- To destroy export records of specific deleted replica datasets:

```
data_sync -site=S1 -f=verify -update -filename=mylist
-classoffile=Dataset
```

The **mylist** file has dataset names in the following format: **dataset123**

- To start synchronization in **pull** mode:

```
data_sync -pull -class=Item -site=S1 -update
-report=report.lst
```

- To force synchronize a list of items specified in a text file in **pull** mode:

```
data_sync -pull -filename="/myhome/itemlist.txt" -force
-update -report=report.lst
```

- To generate a report of which items must be synchronized in **pull** mode:

```
data_sync -pull -filename="/myhome/itemlist.txt" -site=S1
-f=sync -report=report.lst
```

- Include the rendering dataset and **IncrementalChangeElement** items in the synchronized data:

```
data_sync -sync -update -site=site2 -item_id=xxxx
-override_options=opt_incl_rendering:true,opt_include_ice:true
```

- Force updating of the data on the import site even though the last saved dates are the same for the primary and replica:

```
data_sync -sync -update -site=site2 -item_id=xxxx
-session_options=ForceUpdateOnImport:true
```

- To synchronize visualization datasets that are under a replicated item revision that has status:

```
data_sync -OnlyVIS -since=2005-01-01:01:01 -site=Site1 -f=sync
-update -report=report.lst
-include=IMAN_Rendering -include=IMAN_specification
```

- To check if items copied to the **Design_Center** site still exist and delete the IXR from the primary if a copy is no longer there, enter the following command on a single line:

```
data_sync -item_id=* -site=Design_Center -f=verify -update
```

- To delete the IXRs of objects whose replicas do not exist at the remote sites, enter the following command on a single line:

```
data_sync -item_id=* -site=Site1 -f=verify -update
-report=rep.lst
```

- To generate a report of the IXRs, enter the following command on a single line:

```
data_sync -item_id=* -site=Site1 -f=verify -report=rep.lst
```

Both this and the previous example generate reports listing all objects including those that are no longer at the remote site.

- To synchronize all items copied to a site and output a report to a file with newly added components to existing assembly:

```
data_sync -class=Item -site=Site1 -f=sync -update
-include_bom -report=report.rpt
```

- To synchronize any particular item transferred to a replica site and output a report to a file with newly added components to existing assembly:

```
data_sync -u=Tc-admin-user -p=password -g=group -item_id=Item1
-site=Site1 -f=sync -update -include_bom -report=report.rpt
```

- To synchronize all **imanfile** objects copied to a site and output a report to a file:

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

- To force synchronization of **imanfile** objects for all datasets specified in a text file copied to a site and output the report to a file:

```
data_sync -filename=/myhome/datasetlist-for-imanfiles.txt
-classoffile=imanfile -site=Site1 -f=sync -update -report=report.lst
```

- To synchronize files and initiate a checkpoint for three sites:

```
data_sync -f=sync -checkpoint -item_id=item123
-site=Site2 -site=Site3 -site=Site4 -update
```

- To force synchronization of files and initiate a checkpoint for three sites:

```
data_sync -f=sync -update -checkpoint -item_id=item123
-site=Site2 -site=Site3 -site=Site4
```

- To check the status of a given transaction:

```
data_sync -f=status -transaction_id=AhEZaOnRAAMfD
```

- To restart a given transaction for a given site:

```
data_sync -f=sync -transaction_id=AhEZaOnRAAMfD
-restart -site=Site3
```

- To synchronize all 4th Generation Design (4GD) objects copied to a site and output report to a file:

```
data_sync -class=Cpd0DesignElement -site=Site1
-sync -update -report=report.lst
```

- To synchronize specific 4GD objects copied to a site and output report to a file:

```
data_sync -4gd_id=DE000001 -class=Cpd0DesignElement
-site=Site1 -sync -update -report=report.lst
```

- To clean up fast sync transactions prior to specific last process date, list the available transactions to get the last process dates:

```
data_sync -optionset -lt
```

Clean up the transactions:

```
data_sync -optionset -ct -blpd=2012-12-18:20:10:00
```

- To synchronize **MEAppearancePathNode** and **VariantExpression** objects in batch mode with **deferred_batch_size**:

```
data_sync -item_id=<Top Item>
-bo=MEAppearancePathNode,VariantExpression -bs=3000 -update -sync
-include_bom -site=site2 -dbs=3000
```

- To use the **TC_cms_relation_optset_map** preference to include or exclude relations:

1. Add the relation and option set to the **TC_cms_relation_optset_map** preference, for example:

```
IMAN_rendering, opt_rel_rendering
```

- In the PLM XML/TC XML Export Import Administration application, expand the **TransferOptionSet**, click **MultiSiteOptSet**, and add the **opt_rel_rendering** option with the default value set to **false**.
- Expand **ClosureRule**, click **MultiSiteDefaultCR**, and add the following clause:

```
CLASS:WorkspaceObject:CLASS:Dataset:RelationP2S:IMAN_rendering:
SKIP:$opt_rel_rendering==false;
```

This clause states, from any **WorkspaceObject**, find the dataset using **IMAN_rendering** relation, and when the relation is **opt_rel_rendering**, skip the dataset during export. It means the default is to always exclude the **IMAN_rendering** relation for an exported object.

- To include the **IMAN_rendering** in the synchronization using the **data_sync** utility, type:

```
data_sync -include=IMAN_rendering -optionset
```

IMPORTANT NOTES

- When synchronizing items, all item revisions, BOM view revisions, BOM views, forms, and datasets associated with the item will also be synchronized. However, in some cases the item itself is not modified, so the last modification date is not updated and, therefore, cannot be used as the sole basis for synchronization. In most cases, it is necessary to specify all classes associated with an item to guarantee that complete synchronization is accomplished. This means that the command to run the **data_sync** utility should include several class switches, for example:

```
data_sync -class=Item -class=ItemRevision -class=PSBOMViewRevision
```

Note:

If your database contains a large number of replicated items (more than 10,000), you should synchronize one class at a time. When doing so, you should begin with the **Item** class, and then the **ItemRevision** class, followed by the **PSBOMViewRevision** class, and continue down the schema to dataset and forms classes.

- The **PSBOMViewRevision** class must be specified instead of the **PSBOMView** class so that changes to the structure is synchronized.
- When synchronizing an assembly, the **data_sync** utility does not automatically traverse the assembly tree. Rather, it synchronizes each subassembly or component individually on an as-needed basis. If you want the utility to traverse the assembly tree, use the **-include_bom** argument.
- When synchronizing an assembly, **data_sync** transfers new components that are part of the assembly, when sending an assembly with the **-include_bom** argument set to true.

5. Because the **data_sync** utility never involves any transfer of ownership, there is no need to perform export recovery if the utility terminates prematurely.
6. When synchronizing, the utility performs an automatic verification. It checks if the object being synchronized still exists at the remote site prior to synchronizing it. If a replica no longer exists, the utility deletes the corresponding IXR.
7. The **-verbose** argument can be used to analyze the performance of the **data_sync** utility. The **-verbose** argument prints the system times at important stages during the process of synchronization.
8. Siemens Digital Industries Software recommends that you synchronize only one site at a time rather than synchronizing multiple sites in a single run of the **data_sync** utility. This allows you to use the **-same_as_last_export** revision selector that uses the same import/export options used to replicate the item. If you must synchronize multiple sites, create a script that loops through sites but only invokes the **data_sync** utility with only one site at a time.

USING FOLDERS WITH THE DATA_SYNC UTILITY

Folders can be used with the **data_sync** utility, as shown below:

```
data_sync -filename=/tmp/folderlist -classoffile=Folder...
```

If the content of the folder has changed since the last export, if references have been added or removed, the **data_sync** utility updates the remote copy to reflect the current state of the folder.

If no references have been added or removed from a folder since the last export, it is not considered to have been modified. Therefore, if the objects referenced in the folder have changed and need to be updated at the remote site using the **-classoffile=Folder** argument, use the **-force** argument.

GENERATING REPORTS

This example shows how to generate a report called **data_sync.rpt** against the Detroit site:

Enter the following command on a single line:

```
data_sync -class=Item -verify -report=data_sync.rpt -site="Detroit"
```

The results are as follows:

```
Object Date Last Modified Site Date Last Exported Type (Class)
-----
DS_0401_02A 1997-04-03 15:13:50 Detroit 1997-04-03 12:47:45 Text (Dataset)
DS_0401_02A;1 1997-04-03 15:13:40 Detroit 1997-04-03 12:47:48 Text (Dataset)
DS_0401_02A;2 1997-04-03 15:13:43 Detroit 1997-04-03 12:47:52 Text (Dataset)
0320_01/A 1997-03-24 15:34:44 Detroit 1997-03-24 15:33:57 Text (Dataset)
0320_01/A;1 1997-03-24 15:34:33 Detroit 1997-03-24 15:34:00 Text (Dataset)
0320_01/A;2 1997-03-24 15:34:38 Detroit 1997-03-24 15:34:04 Text (Dataset)
```

```
0320_01/A;3 1997-03-24 15:34:42 Detroit 1997-03-24 15:34:06 Text (Dataset)
sueD0324-4;1 1997-03-24 22:04:44 Detroit 1997-03-24 21:57:28 Text (Dataset)
sueD0324-4;2 1997-03-24 22:04:48 Detroit 1997-03-24 21:57:32 Text (Dataset)
```

ERROR CODES

Error code 100228 indicates that a Multi-Site Collaboration file transfer operation has failed. The most likely causes are a network connection failure or an abort (crash) of the IDSM process at the remote site. For the former, retry the operation. For the latter, examine the IDSM system log files at the remote site.

ensure_site_consistency

Allows users to perform corrective actions if the site ownership transaction is interrupted due to a system or network crash or a user-initiated process termination (such as the Windows Task Manager). In cases where legitimate error conditions are encountered (such as lack of transfer privilege or duplicate item IDs), there is no requirement to perform any corrective action; Teamcenter restores the data to consistent states under most non-crash conditions.

Note:

This utility should typically be run only at the exporting site; not at the importing site. The flag that marks an object as requiring this utility is always at the exporting site.

When working with TC XML Multi-Site, use **-f=offline_recovery** in place of **-f=recovery**. When using **-f=offline_recovery**, the **ensure_site_consistency** can be used at both the export and the import site.

SYNTAX

```
ensure_site_consistency [-u=user-id {-p=password | -pf=password-file} -g=group]
-f=clean_all_rec | list_all_rec | offline_recovery | recovery | report | set_owning_site
{ [-item_id=item-id ] | [-key= {keyAttr1=keyVal1 [,keyAttr2=keyVal2],...[,keyAttrN=keyValN]
[-class=class-name] } | [-folder=folder-name] | [-filename=file-name] |
[-itemKeyFile=file-name] | [-search] | [-4gd_id=object-id -class=4gd-class-name] }
[-report=file-name]
[-mode={sst | gms | full | min | auto} ]
[-remote_site=site-name]
[-real_owning_site=site-name]
[-optionset=option-set-name
[-override_options={option1:value1 [,option2:value2],...[,optionN:valueN]}
[-session_options={option1:value1 [,option2:value2],...[,optionN:valueN]}
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-f

Specifies the function to perform. Valid values are:

clean_all_rec Deletes all inconsistent local/replica **IXR**, **ITXR**, and **PAR** records specified in the input that exist for local/replica objects at a local site. This argument deletes workspace objects only. It does not delete **VariantExpression**, **AbsOccData**, or **MEApprPathNode** objects.

Inconsistent local/replica **IXR**, **ITXR**, and **PAR** records occur in the following situations:

- The object is local and the referencing auxiliary objects (**IXR**, **ITXR**, and **PAR**) are replicas.
- The object is replica and the referencing auxiliary objects (**IXR**, **ITXR**, and **PAR**) are replicas.
- The object is replica and the referencing auxiliary objects (**IXR**, **ITXR**, and **PAR**) are replicas.
- At a hub site, the object is local/replica and the referencing auxiliary objects (**IXR**, **ITXR**, and **PAR**) are replicas.

list_all_rec Lists all inconsistent local/replica **IXR**, **ITXR**, and **PAR** records that exist for local/replica objects on a local site. This value must be used with the **report** value.

offline_recovery Performs offline ownership recovery operations. When working with TC XML Multi-Site, use **-f=offline_recovery** in place of **-f=recovery**.

If the item has an SST dataset, use **-f=recovery** to recover the ownership.

recovery Perform recovery operations such as reclaiming site ownership, releasing transfer locks, and removing unwanted export records.

report Generates a list of objects that require recovery. The list is output to a text file identified by the value of the **report** argument. By default, the report contains the Teamcenter Integration Framework transfers (GMS) report followed by the

Synchronous Site Transfer (SST) report. If the **-mode** argument is supplied with this argument, the utility generates a report on the mode specified.

The **clean_all_rec** value can be used with this value.

set_owning_site Sets the owning site of the input objects to the site specified by **-real_owning_site**.

-folder

Specifies a folder that contains items on which to perform corrective action.

The use of a folder is intended for Workspace objects that do not have unique IDs, for example, datasets and forms. This is useful for failed remote checkins of multiple objects where many of the remotely checked-out objects do not have unique IDs, for example, datasets, forms, BVRs, and so forth.

-filename

Specifies a file name that contains a list of items on which to perform corrective action. The file should only contain item IDs. This argument is mutually exclusive with the **-folder**, **-item_id**, and **-search** arguments.

-class (cl)

Specifies the Teamcenter class of the object specified by the **-name** or **-4gd_id** argument. This argument is valid only with the **-name** or, when the **-low_level** argument is specified, with the **-4gd_id** argument. The default class is **Item**.

For organization objects, this argument accepts **Role**, **User**, **Group**, and **Person** classes.

-key

Specifies the keys of the objects on which to perform corrective action. Use the following format:

```
[keyAttr1=keyVal1] [,keyAttr2=keyVal2]...[,keyAttrN=keyValN]
```

To find the key of an object, use the **get_key_string** utility.

-itemKeyFile

Specifies the name of the file containing the keys of the objects on which to perform corrective action.

-search

Finds all the objects that are flagged as requiring corrective action.

This argument is mutually exclusive with the **-folder**, **-filename**, and **-item_id** arguments.

- When used with the **report** function, the utility generates a report on all the objects that are found as requiring corrective actions.

- When used with the **recovery** function, the utility performs corrective actions on the objects that are found as requiring corrective actions.

-4gd_id

Specifies a 4GD object identifier or 4GD object pattern. The **-class=class-name** argument must be specified with the **-4gd_id** argument.

The utility maps the **-4gd_id** argument to the corresponding unique ID of the 4GD class, for example:

```
Class=Cpd0DesignElement, 4gd_id=cpd0design_element_id
```

A 4GD partition object and 4GD subset definition objects do not have a unique 4GD class ID. Therefore, using **-4gd_id** for partition objects or subset definition objects may result corrective action on multiple objects.

To export unique partition objects, use multifield key attributes supplied in the **-key** argument. See *Examples*.

-item_id

Specifies the item ID.

- When used with the **report** function, the utility generates a report on the item specified by *item-ID*.
- When used with the **recovery** function, the utility performs corrective action on the item specified by *item-ID* only if the specified item is flagged as requiring corrective actions.

-report

Specifies the output file path for generating the report. Use this argument with either the **report** function or the **recovery** function.

- When used with the **report** function, the report lists the objects that require corrective action.
- When used with the **recovery** function, the report lists the objects where corrective action was taken.

-mode

Specifies the recovery method type of transfer failures that you want to recover or the type of report to generate when used with the **-report** argument. If you do not specify this argument, the utility uses standard Multi-Site Collaboration transfer failures or generates both a Synchronous Site Transfer (SST) and Teamcenter Integration Framework (GMS) report when used with the **-report** argument. You can specify one of the following valid values:

- **sst**

When used with the **recovery** function, **sst** recovers SST transaction failures.

When used with the **report** argument, the report lists the SST transaction objects that require corrective action.

- **gms**

When used with the **recovery** function, **gms** recovers GMS transaction failures.

When used with the **report** argument, the report lists the GMS transaction objects that require corrective action.

- **full**

When used with the **offline_recovery** function, **full** reimports objects from the metafile and restores site ownership to the local site.

- **min**

When used with the **offline_recovery** function, **min** reads the object UUIDs from the metafile and flips their ownership. A full reimport is not performed. **min** is valid only if the metafile was generated with a transfer of ownership.

- **auto**

When used with the **offline_recovery** function, **auto** restores ownership of the specified item. Objects are not reimported.

-remote_site

Use to convert the replica item to be locally owned (changing the replica to the primary). When used with the **offline_recovery** function and with **-mode=auto**, **-remote_site** specifies the last site for which transfer of ownership was attempted. All objects owned by this site within the given item are owned by the local site. **-remote_site** is not valid with **-real_owning_site**.

-real_owning_site

When used with **f=offline_recovery** with **-mode=auto**, ownership is assigned to the item on the target site specified by **-real_owning_site**. The target site's ISDM must verify that the item exists on the site and that the item on the target site is the item master.

When used with **f=set_owning_site**, ownership is assigned to the item on the site specified by **-real_owning_site** with no ISDM verification.

-optionset

Specifies the name of a transfer option set to use when updating exported objects. The option set must be available at exporting site. The values of the options in the option set are applied to the exported objects.

-override_options

Used with **-optionset** to override options defined in the named option set. Specify options to override with comma-separated name-value pairs. Delimit multiple name-value pairs with semicolons.

-session_options

Used with **-optionset** to override session options defined in the named option set. Specify options to override with comma-separated name-value pairs. Delimit multiple name-value pairs with semicolons.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Setting up a Teamcenter command-line environment *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

EXAMPLES

- Generate a report on all the objects that are flagged as requiring corrective actions:

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

- Generate a report on the item specified by *item_ID*:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=report -item_id=000301 -report=recovery_item.txt
```

- Perform corrective actions on all the objects that are flagged as requiring corrective actions:

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

- Perform corrective actions on the item specified by *item_ID*:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=recovery -item_id=000301 -report=recovery_fixup.txt
```

- Perform corrective actions on a list of item IDs:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=recovery -filename=item_id_list.txt
```

The **item_id_list.txt** file should contain a list of item IDs, one item ID per line.

- Perform corrective actions on all objects under a given uniquely named folder:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=recovery -folder=RecoveryFolderFor26June2007
```

- To restore ownership on an item whose id is **MyCorruptItem**:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=offline_recovery -mode=auto -item_id=MyCorruptItem
-remote_site=Site2
```

- To restore ownership on objects contained in a metafile without reimporting:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=offline_recovery -mode=min -dir=c:\metafile_dir
```

- To reimport objects from the metafile and restore site ownership:

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=offline_recovery -mode=full -dir=c:\metafile_dir
```

- Set the item and all of its dependent objects to be owned by Site2. Collect the dependent objects using the **MultiSiteExpOptSet** option set.

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=set_owning_site -item_id=item001 -real_owning_site=Site2
-optionset=MultiSiteExpOptSet
```

- Set the item and all of its dependent objects to be owned by Site2. Collect the dependent objects using the **MultiSiteExpOptSet** option set and set the option **opt_incl_rendering** with the value **true**.

```
ensure_site_consistency -u=Tc-admin-user -p=password -g=group
-f=set_owning_site -item_id=item001 -real_owning_site=Site2
-optionset=MultiSiteExpOptSet -override_options=opt_incl_rendering:true
```

export_recovery

Recovers and restores exported objects to your database under certain conditions. Occasionally, when you export an object and transfer ownership the object may not be successfully imported at the destination site. This places the object in an undefined state where no one has ownership. The preferred method of correcting this situation is to have the destination site complete the import/export transaction by importing the object into the database from the importing site's **TC_transfer_area** (using interactive object import).

However, if this is not possible, the **export_recovery** utility is used to restore the object to the exporting database from the exporting site's **TC_transfer_area** using the **min** or **full** mode (effectively canceling the export/transfer ownership transaction). If no data is available at either site, recovery can be attempted by running the automode at the exporting site that was the last known owning site.

Use the **export_recovery** utility when an export with transfer of site ownership fails, resulting in objects within an item having inconsistent site ownership. The mode of recovery to use depends on whether there is a valid export directory. The directory must include the **objects.meta** file.

Siemens Digital Industries Software recommends the following order for attempting export recovery procedures; you should try the succeeding procedure only if you cannot perform the previous one or if the previous one fails to restore site ownership:

- If a valid export directory exists (most likely in the **TC_transfer_area** of the exporting or importing site), use either **full** or **min** mode while specifying the valid export directory with the **-dir=** switch. If you attempt to recover at the exporting site, use **min** mode; if you attempt to recover at the importing site, use **full** mode.
- If a valid export directory does not exist, you must attempt recovery from a valid database copy that may be a replica or one with inconsistent site ownership. Use **export_recovery** in **auto** mode. Specify the **-include_bom** switch if appropriate. Specify **-exclude** and/or **-include** switches, if desired.
- If the **auto** mode fails to restore site ownership, perform the manual export recovery procedure:
 1. Define the **TC_EXPORT_COPY=TRUE** environment variable.
 2. Run **item_export** as a Teamcenter administrative user to transfer site ownership to any site.
 3. Run **export_recovery** in **min** mode specifying the directory output in step 2 as the **-dir=** parameter.
 4. If successful, delete the export directory from step 2.

SYNTAX

```
export_recovery [-u=user-id {-p=password | -pf=password-file} -g=group]
               -mode={ full | min | auto | find }
               [-item_id=item-id-to-restore]
```

```
| [-key=[keyAttr1=keyVal1] [,keyAttr2=keyVal2]...[,keyAttrN=keyValN]]
| [-folder=folder-name] | [-filename=file-name]
| [-itemKeyFile=file-name] | [-dir=directory]
[-report=report-file] [-remote_site=last-transfer-site] [-include_bom]
[-real_owning_site=desired-owning-site]
[-exclude=relation-type1 -exclude=relation-type2 ...]
[-include=relation-type3 -include=relation-type4 ...]
[-ignore_am_rules] [-update_lmd] [-bp] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-mode

Specifies the basic mode in which the utility operates. The value of this argument can be one of the following:

full

Restores objects from the export metafile, imports them in to your database and restores ownership to your site.

min

Restores ownership to your site without reimporting data from the metafile. Valid only if the metafile was generated with transfer of ownership.

auto

Restores ownership on the specified item without reimporting. You must specify the **-itemid** argument and either the **-remote_site** or **-real_owning_site** arguments when using this mode.

find

Searches for items with inconsistent site ownership and generates a report.

-dir

Defines the path of the directory containing the exported metafile and the data files. Required only with the **-mode=full** and **-mode=min** arguments.

-item_id

Specifies the ID of the item to process. Wildcards are allowed.

-folder

Defines the name of the Teamcenter folder containing the list of items to process.

-filename

Defines the full path of the file that contains the list of items to process.

-key

Specifies the keys of the items to process. Use the following format:

```
[keyAttr1=keyVal1] [,keyAttr2=keyVal2]...[,keyAttrN=keyValN]
```

To find the key of an object, use the **get_key_string** utility.

-itemKeyFile

Specifies the name of the input file containing the keys to process. The file format is:

```
-key = [keyAttr1=keyVal1][keyAttr2=keyVal2]...
```

```
-key = [keyAttr1=keyVal1][keyAttr2=keyVal2]...
```

```
-key = [keyAttr1=keyVal1][keyAttr2='keyVal2']...
```

-remote_site

Defines the last site for which a transfer of ownership was attempted. This argument is valid only with the **auto** mode.

-report

Specifies the full path of the report file. Valid only with **find** mode.

-real_owning_site

Changes the owning site of specified objects to the site designated. Valid only with the **-mode=auto** argument.

-include_bom

Includes assembly components, if any exist.

-exclude

Excludes the specified relation type and may be given multiple times. The database name (not display name) of the relation type must be used.

-include

Includes the specified relation type and may be given multiple times. The database name (not the display name) of the relation type must be used. Use this argument to force the inclusion of a relation type that is not specified by your **TC_relation_required_on_export** preference.

-ignore_am_rules

Ignores AM rules for recovery purposes.

-update_lmd

Updates the last modified user and date. Valid only with the **-mode=auto** argument.

-bp

Displays best practices information.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

- To use this utility, you must be a user with system administration privileges or be granted authorization by a user with system administration privileges.
- At least one primary mode must be specified.
- Not more than one primary mode can be specified.
- For the **-mode=auto** and **-mode=find** options, exactly one object selection filter (**-itemid**, **-filename**, or **-folder**) must be specified.

EXAMPLES

In each of the following examples, the **-u=user-id** **-p=password** and **-g=group** arguments are assumed:

- To restore ownership on an item with the ID **MyCorruptItem**:

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

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

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

- To reimport objects from the metafile and restore site ownership:

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

- To generate a report of ownership inconsistencies:

```
export_recovery -mode=find -filename=suspect_itemlist.dat  
-report=report.dat
```

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

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

- To restore ownership of an entire assembly:

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

generate_admin_data_compare_report

Generates an interactive report of differences between administration data in two Teamcenter environments. To make the comparison, the utility requires an administration data export package from one environment as a source, and either the local environment or another administration data export package as a target. If data is the same in both the source and the target, the report shows that there are no differences.

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

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

SYNTAX

```
generate_admin_data_compare_report -u=user-ID {-p=password | -pf=password-file} -g=group
-sourcePackage=path-to-source-package
-targetPackage=path-to-target-package | -extractAndCompare
-adminDataTypes={Admin-data1,Admin-data2,...,Admin-dataX | all}
-outputDir=path-to-directory-for-report-files
[-listTypes]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-sourcePackage

Specifies the full path, including the file name, of the source administration data package.

-targetPackage

Specifies the full path, including the file name, of the target administration data package. This argument is mutually exclusive with the **-extractAndCompare** argument.

-extractAndCompare

Extracts the administration data from the current environment and compares it with source package identified by the **-sourcePackage** argument. This argument is mutually exclusive with the **-targetPackage** argument.

-adminDataTypes

Specifies the types of administration data to include in the report. You provide the data types as a comma-separated list (no spaces) or you can specify **all** to get all supported data types in the report.

Tip:

Use the **-listTypes** and **-sourcePackage** arguments to get a list of available administration data types.

If the report contains multiple data types, it includes a where-used table showing where each object is referenced.

-outputDir

Specifies the path to directory where you want the compare report saved. You must specify this argument.

-listTypes

Displays a list of the administration data types at the source that you can compare to the target. You must specify the **-sourcePackage** argument.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

This is a Java utility that, by default, has the maximum Java heap size set to 1024M. For reports that contain a large number of objects, you may need to increase maximum Java heap size to avoid out-of-memory errors or poor performance. If possible, set the maximum heap to at least to 4096M for large reports. You can set this value using the **BMIDE_SCRIPT_ARGS** environment variable, for example:

```
set BMIDE_SCRIPTS_ARGS=-Xmx4096M
```

Note:

Java standards require that no more than 25 percent of total RAM be allocated to virtual memory (VM). If the amount allocated to the Java VM exceeds this percentage, degradation of performance can occur.

FILES

As specified in Log files produced by Teamcenter.

EXAMPLES

- Display a list of the administration data types that you can compare to the source environment package:

```
generate_admin_data_compare_report -u=admin-username -p=admin-password
-g=dba
-sourcePackage=C:\temp\admin_data\SiteA\siteA.zip -listTypes
```

- Generate a comparison report of the preferences at the source and target:

```
generate_admin_data_compare_report -u=admin-username -p=admin-password
-g=dba
-sourcePackage=C:\temp\admin_data\siteA\siteA.zip
-targetPackage=C:\temp\admin_data\siteB\siteB.zip
-adminDataTypes=Preferences
-outputDir=C:\temp\admin_data\compare_sites_A_and_B_preferences
```

- Generate a comparison report of the organization data at the source and target:

```
generate_admin_data_compare_report
-u=admin-username -p=admin-password -g=dba
-sourcePackage=C:\siteA\siteA.zip -extractAndCompare
-adminDataTypes=Organization
-outputDir=C:\temp\admin_data\compare_organization_between_local_and_A
```

generate_admin_data_report

Generates a report showing the specified administration data for the site where you run the utility or for an export package. The export package can be from a remote site.

The report contains HTML pages for the administration data objects, showing their properties with hyperlinks to referenced objects. If an object is referenced by other objects, its HTML page contains a where-used table that indicates the categories and objects that have references to the current object.

The report has a summary showing all the administration data types included in the report and the instances of each element present within the category. The report also has a glossary page with descriptions of the administration data categories and the elements available in each of the categories.

SYNTAX

```
generate_admin_data_report -u=user-ID {-p=password | -pf=password-file}
-g=group
-adminDataTypes=Admin-data1,Admin-data2,...,Admin-dataX | all
[-inputPackage=input-package-path]
-outputDir=path-to-directory-for-report-files
[-listTypes]
[-h]
```

ARGUMENTS

-u

Specifies the user ID. The user must have administrative privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

For more information about managing password files, see *Teamcenter Utilities*.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

-adminDataTypes

Specifies the types of administrative data to include in the compare report. You provide the data types as a comma-separated list (no spaces). You may also specify the **all** value to include all data types defined in the local system or the specified input package.

Tip:

Use the **-listTypes** argument to get a list of available administration data types.

If the report contains multiple data types, it includes a where used table showing where each object is referenced.

-inputPackage

Specifies the full path, including the file name, of the export administration data package from the site for which the report is generated. If you do not specify this argument, the utility generates a report for the local site.

-outputDir=

Specifies the path to directory where you want the report saved. You must specify this argument.

-listTypes

Displays a list of the available administration data types that you can include in the report.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment.

This is a Java utility that, by default, has the maximum Java heap size set to 1024M. For reports that contain a large number of objects, you may need to increase maximum Java heap size to avoid out-of-memory errors or poor performance. If possible, set the maximum heap to at least to 4096M for large reports. You can set this value using the **BMIDE_SCRIPT_ARGS** environment variable, for example:

```
set BMIDE_SCRIPT_ARGS=-Xmx4096M
```

Note:

Java standards require that no more than 25 percent of total RAM be allocated to virtual memory (VM). If the amount allocated to the Java VM exceeds this percentage, degradation of performance can occur.

FILES

As specified in Log files produced by Teamcenter.

EXAMPLES

- Generate a list of the administration data types that you can export:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba  
-listTypes
```

- Generate a report containing the preferences and their values at the local site:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba  
-adminDataTypes=Preferences  
-outputDir=C:\temp\admin_data\siteA\preferences_report
```

- Generate a report containing the Access Manager and Organization administration data from an export package of a remote site:

```
generate_admin_data_report -u=admin-username -p=admin-password -g=dba  
-adminDataTypes=AccessManager,Organization  
-inputPackage=C:\temp\admin_data\siteB\siteB.zip  
-outputDir=C:\temp\admin_data\siteB\am_and_organization_report
```

Tip:

The export package is generated using the **admin_data_export** utility.

import_file

Imports files into the Teamcenter database according to a set of user-specified arguments. These arguments supply user identification information, dataset information, and (optionally) item information to be associated with the imported file. The arguments may be specified on the command line to import a single data file or in a file to import multiple data files (bulk import).

Depending on the arguments, each data file is copied (an **ImanFile** object is created), a dataset is created (or modified), and if specified, an item is created or modified to contain the dataset. In the absence of a specified item, the dataset is placed in the user's **Newstuff** folder.

The **import_file** utility does not support the creation of custom item types.

SYNTAX

```
import_file [-u=user-id {-p=password | -pf=password-file} -g=group]
-f=file-name | -i=file-name [-vb] [-log=file-name] -type=datasettype -d=dataset-name
-ref=named-reference [-de={n | e | a | r}] [-item=item-id | -itemkey=key-id]
[-itemRevUid=item-revision-uid]
[-relationType=relation-type] [-use_ds_attached_to_rev_only]
[-revision=item-rev-num] [-ie={n | y}] [-desc=string]
[-v=volume-name] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-f

Imports a single file into Teamcenter. The full path must be provided if the file does not reside in the current working directory. The **-f** and **-i** arguments are mutually exclusive. See example 1.

-i

Imports multiple files into Teamcenter using a specified import file. The full path must be provided if the file does not reside in the current working directory. The **-f** and **-i** arguments are mutually exclusive. See example 2.

-vb

Runs utility in verbose mode. Displays maximum amount of information. Nonverbose sessions only display error messages.

-log

Creates a log of items and datasets created.

-type

Defines the dataset type in Teamcenter, for example, **Text** or **UGPART** datasets. **-type** is case sensitive and must match the dataset type name as defined in the Teamcenter data model.

-d

Specifies the name of the dataset into which the file is imported.

-ref

Specifies the type of named reference associated with the file. The value specified by this argument may or may not be identical to the value specified by the **-type** argument.

For example, **Text** or **UGPART** type datasets have named references of **Text** and **UGPART**, respectively. However, **DirectModel** type datasets have a **JTPART** named reference.

Each dataset type defines one or more named references to be associated with it. See restriction numbers 1 and 2.

-de

Indicates that a dataset exists. Used when a dataset of the same name already exists.

=n

Specifies that a new dataset be added even if one with the same name exists. If it does exist, it is added to the same item folder. If it does not exist, it is placed in the new item folder or the user's **Newstuff** folder.

=e

Specifies that a new dataset be added if the dataset name specified by the **-d** argument is not used by any existing dataset. The utility displays an error message if you supply this argument and the dataset name specified by the **-d** argument already exists in Teamcenter.

=a

Specifies that the imported file be added as a named reference to the existing dataset. When this is done, a new dataset version is created. This version contains the additional imported named reference file.

=r

Specifies that a new dataset revision be created and the existing named reference be replaced with the new one. This option generates an error if the dataset has no existing named reference.

-item

Specifies the name of the item containing the dataset that references the imported file.

-itemkey

Specifies the key of the object. You can use the **-item** argument or the **-itemkey** argument.

To find the key of an object, use the **get_key_string** utility.

-itemRevUid

Specifies the 14 character UID string of the item revision object where the file and dataset are to be attached.

-relationType

Specifies a relation to use when the **IMAN_specification** relation is not appropriate. If the **-relationType** argument is not specified, the **IMAN_specification** relation is used.

-use_ds_attached_to_rev_only

Specifies a dataset based on its name and type, but it considers datasets attached to the specified item and item revision only. The item and item revision are specified by the **-item** and **-revision** parameters, respectively.

This prevents the utility from referring to datasets with the same name and type, but that are unattached or are attached to another item or revision.

If you want to revise the existing dataset instead of creating a new one, the **-use_ds_attached_to_rev_only** parameter is particularly useful when used along with the **de=r** argument.

-revision

Specifies the item revision number and revision ID. See restriction number 3.

-ie

Specifies behavior if the item already exists.

=n

Specifies that the dataset will not be added if the item already exists.

=y

Specifies that the dataset may be added if the item already exists. If the item exists, but the item revision does not, an item revision is created.

-desc

Specifies a user-defined text description of an item that is created by the import function. If the **import_file** utility is creating a new revision of an existing item, this is the description of the item revision.

-v

Specifies the full path of the Teamcenter volume where the imported file is placed.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

1. To create a dataset in Teamcenter, the user must specify the dataset type and the named reference.
2. When importing a file as a dataset, you must specify the named reference using the **-ref** argument.
3. When importing a file into an item revision, you must specify the revision; otherwise, an error message displays indicating a missing revision.

EXAMPLES

- To import a single operating system file, **bike.dat**, into Teamcenter as a **UGPART** dataset named **my_bike_dataset**, enter the following on a single line:

```
$TC_ROOT/bin/import_file -user=user-id -p=password -g=group -f=bike.dat
-type=UGPART -d=my_bike_dataset -ref=UGPART
```

- To import multiple operating system files into Teamcenter, first create an input file that contains the following information:

```
-f=bike1.dat -d=my_bike1_dataset -type=UGPART -ref=UGPART
-f=bike2.dat -d=my_bike2_dataset -type=UGPART -ref=UGPART
```

```
.
-f=binkeN.dat -d=my_bikeN_dataset -type=UGPART -ref=UGPART
```

- Run the **import_file** utility using the input file from example 2, entering the following command on a single line:

```
$TC_ROOT/bin/import_file -user=user-id -password=password -group=group
-i=input-file
```

- Import the **d:\some_file.jt** file:

```
%TC_ROOT%\bin\import_file -user=user-id -p=password -group=group
-f=d:\some_file.jt -type=DirectModel -d=my_jt_file_dataset
-ref=JTPART
```

- Import the **d:\WordDoc.doc** file:

```
%TC_ROOT%\bin\import_file -user=user-id -p=password -group=group
-f=d:\WordDoc.doc -type=MSWord -d=my_word_dataset -ref=word
```

- Import the **d:\ExcelFileTest.xls** file:

```
%TC_ROOT%\bin\import_file -user=user-id -p=password -group=group
-f=d:\ExcelFileTest.xls -type=MSExcel -d=my_excel_dataset
-ref=excel
```

- Import the **d:\myfile.txt** file:

```
%TC_ROOT%\bin\import_file -user=user-id -p=password -group=group
-f=d:\myfile.txt -type=Text -d=my_text_file_dataset -ref=Text
```

item_export

Exports a single item or multiple items in batch mode. It is the companion to the **item_import** utility. This utility supports part family templates and members and works with the **TC_relation_required_on_export** and **TC_relation_required_on_transfer** preferences.

SYNTAX

```
item_export [-u=user-id {-p=password | -pf=password-file} -g=group]
-dir=directory
{-item=item-id | -keykey-id [-rev=revision-selector] | -filename=input-file
| -itemKeyFile=file-name}
{-owning_site=site-name | -target_site=site-name1, site-name2, ...}
[-exclude=relation-type1 -exclude=relation-type2...]
[-include=relation-type1 -include=relation-type2...]
[-reason=export-reason] [-latest_ds_version] [-include_bom]
[-batch_objects=list-of-deferred-classes]
[-batch_file=file-name-listing-deferred-classes]
[-deferred_batch_size=batch-size-for-deferred-objects]
[-preview] [-report=file-name] [-continue_on_error]
[-xfr_top_lvl_only] [-xfr_top_asm_only] [-xcl_files]
[-status=release-status] [-exclude_folder_contents]
[-classoffile=class-name] [-separator=separator-character]
[-dont_exclude_protected] [-email=email-address] [-script=script-name]
[revision-selector] [-include_bc] [-include_supercedures] [-v] [-h]
```

ARGUMENTS

Entries in parentheses are accepted abbreviations for arguments.

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-dir

Specifies the full path of the directory where the metafile and data files are stored.

-item

Specifies the ID of the item to be exported. Valid only if no input file is specified using the **-i** argument.

-key

Specifies the key of the object. You can use the **-item** argument or the **-key** argument.

To find the key of an object, use the **get_key_string** utility.

-rev

Identifies the revision to be exported. This can be the revision ID or one of the following keywords:

LATEST
LATEST_WORKING
LATEST_RELEASED
LATEST_WORKING_OR_RELEASED
USE_STATUS

If the **USE_STATUS** keyword is given, you must specify a release status using the **-status** argument. This is valid only when you are not transferring site ownership.

If used with an input file (**-i** argument), the revision keyword is used for every item in the input file. Keywords cannot be specified using the command line; however, you can use revision selectors at the command line as discussed below.

Note:

The revision ID cannot be specified when using an input file.

-filename (fn)

Specifies the name of an input file that contains the list of item IDs to be exported.

The format of the text file must contain the **-item=** prefix to each item ID. For example, to export item IDs **002259**, **002260**, and **002261**, the input file contains the following entries:

```
-item=002259
-item=002260
-item=002261
```

-keyFileName

Specifies the name of the input file containing the keys to be exported. The file format is:

```
-key = [keyAttr1=keyVal1][keyAttr2=keyVal2]...
-key = [keyAttr1=keyVal1][keyAttr2=keyVal2]...
-key = [keyAttr1=keyVal1][keyAttr2='keyVal2']...
```

-classoffile (cof)

Specifies the class of objects contained in the input file. If no class is specified, the default class is **Item**. Valid only with the **-filename** argument.

-separator (sep)

Specifies the character to separate the item and revision IDs in the file. The default is **/**.

-target_site (ts)

Specifies the export target site or sites. If more than one site is specified, sites must be separated by a comma and the entire string must be enclosed in quotes. Either the **-target_site** or **-owning_site** argument is required.

-owning_site (os)

Specifies the site to which ownership is transferred. Either the **-target_site** or **-owning_site** argument is required.

-exclude (exc)

Specifies the relation type to be excluded. This argument may be given multiple times, and the database name (not the display name) of the relation type must be specified. You cannot exclude the **IMAN_master_form** and **TC_ic_intent_rtype** relation types with or without ownership transfer. Also, you cannot exclude the **IMAN_RES_audit** with ownership transfer.

-include (inc)

Specifies the relation type to be included. This argument may be given multiple times, and the database name (not the display name) of the relation type must be specified. You cannot include the **IMAN_RES_checkout** and **IMAN_based_on** relation types.

-exclude_folder_contents (efc)

Excludes the contents of a folder. Intended for use with NX Part families where family members are stored in a folder that is attached to the item.

-dont_exclude_protected (dxp)

Does not exclude export-protected objects. If set, any export-protected object within an item prevents the export of the entire item.

-reason (rea)

Specifies the reason for exporting to sites. Up to 240 characters.

-latest_ds_version (ldv)

Exports only the latest version of datasets; default is to export all versions. Valid only when site ownership is not being transferred.

-include_bom (bom)

Exports all components if the given item is an assembly.

-preview (pre)

Performs an export dry run and generates a report to the file specified by the **-report** argument. If the **-report** argument is not specified, the report is output to the screen.

-report (rep)

Outputs preview or completion reports to the specified file. If no report file name is specified, the report is output to the screen.

-continue_on_error (con)

Continues the export operation even if errors are detected on optional objects. Optional objects are attachments other than requirement, specification, or primary form objects.

-xfr_top_lvl_only

Only transfers ownership on top-level items specified in the input file.

-xfr_top_asm_only

Transfers ownership only on the top-level assembly items, as specified in the input.

-xcl_files

Excludes export of files in datasets.

-include_pfmembers

Identifies the related part family members to be exported when handling part family templates.

-include_pftemplates

Identifies the related part family template to be exported when handling part family members.

-pf_bom_treatment

Identifies the part family objects associated with the assemblies to be exported. The argument must be used in conjunction with the **-include_bom** argument. Valid options are:

-members

Includes part family member components present in the assembly.

-templates

Includes part family template rather than part family member components.

-all

Includes both the part family member components and templates.

-none

Includes neither the part family member components nor the templates.

-status (stat)

Specifies the release status type to use for selecting the item revision to be exported.

-include_bc

Exports the change revision along with the **BOMChange** objects associated with the affected assemblies of the change revision.

-include_supercedures

Exports the change revision along with the supercedures associated with the **BOMChange** objects.

-email

Specifies the email address to which the export report is sent.

The default address is stored in the Teamcenter user account.

-script

Specifies the name of the script in the **TC_BIN** directory that is executed after a successful export. If a script is already defined by the **TC_post_export_scriptTC_post_export_script** preference, the specified script overrides the preference entry.

-revision_selector

Determines which item revision is exported with the item. The valid selectors are as follows:

latest_revision (lt)	Exports the latest revision only, regardless of whether it is a working or released revision.
latest_working (lw)	Exports the latest working revision only.
latest_released (lr)	Exports the latest released revision only with any release status.
latest_working_or_any (lwoa)	Exports the latest working revision. If no working revision exists, it exports the latest released revision with any release status.
status (stat)	Specifies the release status to be exported.

If no revision selector is given, all revisions are exported.

Note:

Revision selectors should be capitalized only when used with the **-rev= switch** and should be in lower case when used as a switch.

-v

Runs utility in verbose mode to display maximum amount of information. Typically, nonverbose utility sessions only display error messages.

-batch_objects (bo)

Specifies a list of comma-separated deferred classes. If you use this argument with the **preview** argument, only nondeferred objects with the number of deferred objects appear in the report.

-batch_file (bof)

Specifies the file name of a text file containing a list of deferred classes.

-deferred_batch_size (dbs)

Specifies the number of objects per batch; a new process is created per batch. The default value is **1000**. This value must be a positive integer. Use this argument to process thousands of objects to avoid memory and disk shortage problems.

The following classes are supported for deferred objects:

- **Dataset**
- **Folder**
- **Form**
- **ImanRelation**
- **MEAppearancePathNode**
- **NamedVariantExpression**
- **PSOccurrence**
- **VariantExpression**
- **VariantExpressionBlock**

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

1. The **-item** argument is mutually exclusive with the **-i** and **-filename** arguments.
2. Either the **-target_site** or **-owning_site** argument must be specified and must not be a local site.
3. It is the responsibility of the user exporting objects to inform the system administrator which directories need to be exported and to which site.
4. It is the responsibility of the system administrator to set up the list of other sites which are known to the local site.
5. It is the responsibility of the system administrator to send directories of the exported objects to the receiving sites, users, volumes, and other systems.
6. Administration object types cannot be exported.

EXAMPLES

To restart a checkpoint transaction that failed during import:

```
item_export -transaction_id=AjEzZaOnRAAAffD -restart
```

item_import

Imports multiple items (in batch mode) into the Teamcenter database. It is the companion to the [item_export](#) utility. This utility supports part family templates and members.

SYNTAX

```
item_import [-u=user-id {-p=password | -pf=password-file} -g=group]
  -dir=directory
  [-folder=folder-name] [-preview] [-report=file-name] [-filename=file-name]
  [-classoffile=class-name] [-list_metafile] [-include_pfmembers=part-family-members]
  [-include_pftemplate=part-family-templates]
  [-part_family_bom_treatment={members | templates | all | none}]
  [-script=pre-import-script] [-email=email-address]
  [-parallel=number-of-parallel-processes] [-continue_on_error] [-verbose]
  [-transaction_id=tid] [-restart] [-h]
```

ARGUMENTS

Entries in parentheses are accepted abbreviations for arguments.

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-dir

Specifies the path name to the directory containing the metafile and the data files to be imported.

-folder (fol)

Specifies the destination folder in which imported items are placed. If the folder does not exist or the argument is not supplied, the imported items are placed in the user's **Newstuff** folder.

-preview (pre)

Performs an import dry run and generates a dry run report to the file specified by the **-report** argument. If the **-report** argument is not specified, the report is output to the screen.

-report (rep)

Outputs a preview or completion report to the specified report file. If no report file name is specified, the report is output to the screen.

-filename (fn)

Specifies the name of the text file listing objects for selective import, one name per line.

-classoffile (cof)

Specifies the class of objects contained in the input file. If not specified, the default class is **Item**.

-list_metafile (lm)

Lists only the contents of the metafile; does not import objects.

-include_pfmembers

Identifies the related part family members to be imported when handling part family templates.

-include_pftemplate

Identifies the related part family template to be imported when handling part family members.

-part_family_bom_treatment

Identifies the part family objects associated with the assemblies to be imported. The argument must be used in conjunction with the **-include_bom** argument. Valid options are:

-members

Includes part family member components present in the assembly.

-templates

Includes part family template rather than part family member components.

-all

Includes both the part family member components and templates.

-none

Includes neither the part family member components nor the templates.

-script

Specifies the name of the script to be executed prior to import. If a script is defined by the **TC_post_export_script** preference, this argument overrides the preference setting. If specified as **NONE**, the script defined in the preference file is executed.

-email

Sends email to the user at the specified email address after completion. If no email address is specified, the email address in the user's Teamcenter user profile is used.

-parallel (par)

Specifies the number of processes to be started automatically. If this argument is not specified, the system imports the deferred objects with a sequential process.

-continue_on_error (con)

Specifies that the import operation proceeds when an error has occurred on an optional object, such as a reference or manifestation attachment.

-verbose (v)

Runs the utility in verbose mode to display the maximum amount of information. Typically, nonverbose utility sessions only display error messages.

-transaction_id (trid)

Specifies the transaction ID for a given checkpoint-related operation.

-restart (rs)

Restarts a given transaction at the point of failure.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

EXAMPLES

To restart a transaction at a site where a failure occurred:

```
item_import -transaction_id=transaction-id -restart
```

item_relink

Replaces the external references for a duplicate item and its corresponding replica in a Multi-Site Collaboration environment. The **item_relink** utility works in conjunction with the **item_rename** utility.

The **Bypass** option enables or disables special bypass object protections for Teamcenter administrators, allowing you to freely access any object in the database to perform maintenance. When running this utility, you must use the **Bypass** option, the user ID must be **infodba**, and the OS user account must have read-access to the NX part files to comply with the rules stated in **Bypass UG Part File Verification**.

Caution:

The **item_relink** utility is used only with Multi-Site Collaboration to process production data. Siemens Digital Industries Software recommends that a full backup of your database be performed before running this utility. This allows you to restore the database if the data becomes corrupted.

- **Naming pattern**

Because the **item_relink** utility works in conjunction with the **item_rename** utility, the same naming pattern must be used in both utilities.

- **Bypass NX part file verification**

Each NX part file that is attached to a duplicate is checked against the corresponding NX part file that is attached to the replica. The **item_relink** utility compares the UID strings in the NX part files. The **ug_inspect** utility retrieves the UID strings from the part files. Therefore, it is very important to run the **item_relink** utility using the OS account that has read access to the part files. Usually, the Teamcenter user account, such as **infodba**, is used to run the utility. If UIDs are not the same, the relink process for the duplicate fails. If you are confident about the part files being reconciled, you can use the **-bypass_ugpart** command line argument to bypass this check. The **-bypass_ugpart** argument is ignored if the **item_relink** utility is run in verify mode.

- **Matching criteria**

To find the corresponding replica, construct the replica item ID based on the duplicate item ID and renaming pattern and then search the database for the replica.

To find the corresponding item revision, match the revision ID.

To find the corresponding BOM view, match the view type name.

To find the corresponding BOM view revision, match the view type name.

To find the corresponding secondary object, match the object name, object type, and relation type.

- **Matching results**

For each object that is attached to a duplicate item or duplicate item revision, if more than one object with the same object name, object type, and relation type are found in its corresponding replica item or replica item revision, the first occurrence is used.

If objects attached to a duplicate do not have corresponding objects found in replica, use the **-verify** switch to generate a report that lists any discrepancies. In this case, perform a detailed examination and make the necessary corrections and/or ownership change for the duplicate. If any discrepancies are detected during the relink process, the duplicate is not replaced. Instead, the duplicate is placed in the exception folder. An error message is logged on the report file for review.

- **Exception**

If unexpected Teamcenter internal errors occur or the duplicate contains objects not found in its corresponding replica, the utility stops processing the duplicate that has a problem, logs an error message to the report file, and then processes the next duplicate in the replacement folder. All duplicates that are not reconciled are placed in the **Item_ID_ConsolidationEXP** exception folder so you can further examine these duplicates.

SYNTAX

```
item_relink [-u=user-id {-p=password | -pf=password-file} -g=group]
-replace=folder-name -refile=folder-name
-update | -verify
[-prefix=prefix-removed-from-item-id | -suffix=suffix-removed-from-item-id | -f=file-name]
[-bypass_ugpart ]
[-ignore_attachments]
[-relink_to_latest_rev]
-report | -report=file-name
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-replace

Specifies the name of the folder that holds items that are currently duplicates but should be replicas. This must be the same folder used by the **item_rename** utility.

-verify

Requests verification of compatibility between duplicates and replicas.

-update

Performs the link replacement.

-prefix

Specifies the prefix removed from the item ID of duplicates to form the new item IDs for replicas. This must be the same prefix used by the **item_rename** utility. See restriction 5.

-suffix

Specifies the suffix removed from the item ID of duplicates to form the new item IDs for replicas. This must be the same suffix used by the **item_rename** utility. See restriction 5.

-f

Specifies the file containing item ID cross reference records. The cross reference contains the duplicate item ID and the renamed duplicate item ID for each duplicate item ID. This data is contained in a single 80-byte line in the file. The **item_rename** utility also uses this file. See restriction 5.

-bypass_ugpart

Indicates whether NX part files are verified. If this switch is specified, no verification is performed. This switch is ignored if the **-verify** argument is specified.

-ignore_attachments

Prohibits linking of secondary objects (attachments). Use this argument when the replacement item already has links to all required attachments.

-relink_to_latest_rev

Links all revisions of a duplicate item to latest revision of the replica item. Ignores secondary objects (attachments) to avoid incorrect attachments. This argument cannot be specified for items with multiple views.

-report

Generates a report. Outputs the report to standard output if the file name is not supplied.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The following restrictions must be understood and adhered to when using the **item_relink** utility:

1. Must be run with the **Bypass** option, and the user ID must be **infodba**.
2. The OS user account must have read-access to the NX part files to comply with the rules stated in **Bypass UG Part File Verification**.
3. The **-replace** and **-refile** arguments must be supplied.
4. Either the **-rename** or **-verify** arguments must be supplied.
5. A default naming pattern is used if the **-prefix**, **-suffix**, or **-f** argument is not supplied.

EXAMPLES

The best practice is to run the **item_relink** utility with the **-verify** argument to do a comparison to find discrepancies between duplicates and replicas. If any exist, examine the discrepancies and make the necessary corrections. To ensure data integrity, Multi-Site Collaboration imposes strict rules on object replication. One of these rules is that only the primary object can be modified. The replicated object must never be checked out for modification or submitted for release. Therefore, if the duplicates contain objects that have no corresponding replicas, the relink process for these duplicates is not performed. However, if the replicated objects have increased with more attachments, the duplicates are overwritten.

- To verify the items in the replacement folder and generate a report called **relink.rpt**, enter the following command on a single line. The naming pattern must be the same as that used by the **item_rename** utility.

```
Item_relink -u=infodba -p=infodba -replace=replacement  
-refile=assm_refile -prefix=AAA -verify -report=relink.rpt
```

- After generating a replacement report, you may need to correct duplicates or change ownership. To replace the links, enter the following command on a single line:

```
Item_relink -u=infodba -p=infodba -replace=replacement  
-refile=assm_refile -prefix=AAA -update -report=relink.rpt
```

item_rename

Changes the item IDs for duplicate part numbers in a naming pattern in a Multi-Site Collaboration environment. The **item_rename** utility works in conjunction with the **item_relink** utility. The main reason for renaming duplicates is to avoid a naming conflict while bringing in copies of the primary data that was previously created.

The **Bypass** option enables or disables special bypass object protections for Teamcenter administrators, allowing you to freely access any object in the database to perform maintenance. When running this utility, you must use the **Bypass** option and the user ID must be **infodba**.

- **Naming pattern**

You can use the **-prefix**, **-suffix**, or **-f** arguments to embed a renaming pattern for the duplicate data objects. If these arguments are not used, the system applies a default naming pattern. The default naming pattern adds the **DUP_** prefix to the duplicate item IDs. For example, if the duplicate item ID is **ABC123**, after the **item_rename** utility runs the duplicate item ID is **DUP_ABC123**.

The **-prefix** and **-suffix** switches enable you to add character strings to the item IDs to form new item IDs.

The **-f** switch supplies a file that contains a list of item ID cross-references. If the **-f** argument is specified, the system ignores the **-prefix** and **-suffix** switches.

- **Cross-reference file format**

The **-f** switch generates a file that contains a list of item ID cross-references, specifically the duplicate item ID and the renamed duplicate item ID. Each set of item IDs is contained in a single 80-byte line. The duplicate item ID precedes the renamed duplicate item ID. The duplicate item ID and the renamed duplicate item ID must be separated by at least one blank space, although more are allowed. Leading blanks may appear before the duplicate item ID or padding blanks may appear after the renamed duplicate item ID.

The system administrator manually creates the cross-reference file. The system administrator must know how to match the item ID replicas and the item ID duplicates.

- **Exception**

If any unexpected Teamcenter internal errors occur, the utility stops processing the duplicate that has a problem, logs an error message to the report file, and then processes the next duplicate in the replacement folder.

The **item_rename** utility is used only with Multi-Site Collaboration.

SYNTAX

```
item_rename [-u=user-id {-p=password | -pf=password-file} -g=group]
-replace=folder-name -rename | -verify [-prefix= prefix-added-to-item-id | -suffix= suffix-added-to-item-id | -f=file-name] -report=file-name [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-replace

Specifies the name of the folder that holds the items that are currently duplicates but that should be replicas.

-prefix

Specifies the prefix added to the item ID of duplicates to form the new item IDs.

-suffix

Specifies the suffix added to the item ID of duplicates to form the new item IDs.

-f

Specifies the file containing item ID cross-reference records. The cross-reference is comprised of the duplicate item ID and the renamed duplicate item ID for each duplicate item ID. This data is contained in a single 80-byte line in the file. The **item_relink** utility also uses this file.

-verify

Requests verification of the existence of renamed items.

-rename

Performs the rename function.

-report

Generates a report and outputs it to standard output if the file name is not supplied.

-h

Displays help for this utility.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The following restrictions must be understood and adhered to when using the **item_rename** utility:

1. Must use the **Bypass** option, and the user ID must be **infodba**.
2. The **-replace** argument must be supplied.
3. Either the **-rename** or **-verify** argument must be supplied.
4. A default naming pattern is used if either the **-prefix**, **-suffix**, or **-f** arguments are not supplied.

EXAMPLES

The best practice is to run **item_rename** with the **-verify** switch to do a quick search for any objects with the chosen naming pattern. If any exist, choose a different naming pattern for all objects.

- Enter the following command on a single line to verify the items in the replacement folder and generate a report called **rename.rpt**. Assume that the naming pattern adds the prefix **AAA** to the item ID.

```
Item_rename -u=infodba -p=infodba -replace=replacement  
-prefix=AAA -verify -report=rename.rpt
```

If any items in the database have the same item ID as the chosen naming pattern, error messages beginning with *****ERROR** are logged on the **rename.rpt** file.

- Change the naming pattern and run the **item_rename** utility again. Otherwise, use the same command line in step 1, and replace the **-verify** argument with the **-rename** argument to rename the items.

plm_report_extract

DESCRIPTION

Extracts the persistent data from a Teamcenter site. This extract can be used to perform consistency and constraint analysis with similar extracts from other Teamcenter sites using the **plm_report_consistency_anlayis** and **plm_report_constraint_analysis** utilities. The extract is in binary format and is not human readable.

SYNTAX

```
plm_report_extract -u=user-name {-p=password | -pf=file} [-g=group]
-attributes_filefile {-item_id=single item-id or pattern | -name=wso-name [-class=class] |
-inputfile=file [-class=class] } -output_file=file [-exclude=file] [-optionset=option-set] [-extract_ods]
[-verbose] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-class

Specifies the object's Teamcenter class, as specified by the name argument. This argument is only valid with the **-name** or **-input_file** arguments. The default class is **Item**.

-name

Specifies the name argument of a single workspace object to be processed. The default class is **Item**. The **-class** argument is required when the desired object class is not **Item**. If you supply **-class=Item** with this argument, the **-name** argument is treated as the **item_id** attribute.

-attributes_file

Specifies the absolute path to the attributes file, which lists the attributes to be extracted. A default set of attributes is supplied.

The **default_attributes_file.txt** is present in **TC_DATA** location.

-input_file

Specifies the filename, with absolute path, to the input file containing the list of objects to process. By default, file entries are treated as IDs for **Item** and **ItemRevision** objects.

If the **-class** argument is used to specify workspace objects other than **Item** and **ItemRevision**, the file entries are treated as names. Each entry should be separated by a comma or a new line character.

-item_id

Specifies the item ID or template (pattern) of items to process. *GMO000** indicates a pattern to include all item IDs that start with *GMO000*. Mutually exclusive with **-inputfile**, **-name** and **-class** arguments.

-output_file

Specifies the file name and absolute path where the extracted output file is created.

-extract_ods

Extracts associated ODS information.

-optionset

Specifies the **OptionSet** used for traversal. The default is **SiteConsolidationDefault**.

Caution:

Before you run this utility with the lightweight closure rules (**SiteConsolidationLW**), ensure the **internalclosurerule** option is set to **false**. This is the default value for this option; however, if it is set to **true**, it can cause the utility to go into an infinite loop.

-verbose

Displays additional progress information.

-exclude

Specifies the full path and filename containing the list of objects to be excluded.

-h

Displays help information.

ATTRIBUTE MAP

A configuration mechanism lets you specify additional attributes to be extracted by **plm_report_extract**. The attributes map file can be specified as input with the **-attributes_file** command line argument. The format of this file is:

```
Column Name 1:Class Name1,attr1;Class Name2,attr2
Column Name 2:Class Name1,attr1;Class Name2,attr2
```

This is an example configuration:

```
Owning Site:POM_Object,owning_site
last_mod_date:POM_application_object,last_mod_date
last_mod_user:POM_application_object,last_mod_user
Name:WorkspaceObject,object_name;Item,item_id;ImanFile,original_file_name
```

When you customize the attribute column, you can refer to the Teamcenter Business Modeler IDE to find class attribute names defined in the database. Note that the runtime properties listed in the Business Modeler IDE are not applicable here; only persistent class attributes can be specified.

Caution:

Do not edit the existing attribute file. Add additional entries if required.

ENVIRONMENT

As specified in the *Teamcenter Utilities*. In addition, the log file is created in the local directory specified by either the **TMP** or **TEMP** environment variable.

RESTRICTIONS

None.

EXAMPLES

- To extract data for multiple items based on their item ID pattern:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -item_id=GM0000*  
-output_file=d:\data\TestMEP.txt -attributes_file=D:\Config\mapfile.txt
```

- To extract data for objects specified in an input file:

```
plm_report_extract -u=tc-admin-user -p=password  
-g=group -inputfile=D:\Input\list_of_item_ids.txt  
-output_file=d:\data\TestMEP.txt -attributes_file=D:\Config\mapfile.txt
```

plm_report_consistency_analysis

DESCRIPTION

Compares the results of multiple site extracts based on the extracts produced by the **plm_report_extract** utility, analyzes the comparison, identifies Multi-Site-related issues, and suggests fixes for the issues found.

The **plm_report_consistency_analysis** utility can generate a full text version of the binary file generated by the **plm_report_extract** utility.

The report contains a header in the following format to allow you to plan and track incremental site consolidation by reporting the sites analyzed and the files that are used:

```
Inconsistency Analysis Report Created at [2013-02-14 11:52:51]
Site ID=[100001] Site Name=[tc101ms1] - Extraction Date: [2013-02-12
18:42:33]
Extract File path=[E:\tasks\PR\6827629\extract\const_000001_src.xml]
Site ID=[100001] Site Name=[tc101ms1] - Extraction Date: [2013-02-12
18:42:54]
Extract File path=[E:\tasks\PR\6827629\extract\const_000002_src.xml]
Site ID=[100001] Site Name=[tc101ms1] - Extraction Date: [2013-02-12
17:21:40]
Extract File path=[E:\tasks\PR\6827629\extract\remote_object_001_src.xml]
```

Reports generated by this utility can be imported into other applications such as Microsoft Excel for further processing. The default delimiter, pipe character (vertical bar |), should be used when importing into Microsoft Excel.

Because this utility uses the pipe character as a delimiter for creating the uncompressed report, if an object's descriptive data, such as the **Description** attribute, contains pipe characters, the uncompressed output is distorted. This can also happen if newline characters are present in the descriptive data.

SYNTAX

```
plm_report_consistency_analysis -u=user-name {-p=password | -pf=file}
[-g=group] -report_type=consistency_analysis | uncompress -input_dir=directory
-report_dir=directory [-sites=site1,site2,..] [-suggest_no_fixes] [-extract_ods] [-verbose] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-report_type

Specifies either **consistency_analysis** or **uncompress**.

consistency_analysis

Performs consistency analysis for extracts generated by the **plm_report_extract** utility, and report the cross-site and internal site inconsistencies for the sites whose extracts are being analyzed.

uncompress

Uncompresses the extract file to human readable format. If this option is used all other options are ignored.

-input_dir

Specifies the directory containing all the extract files.

All the extract files should be generated using the same attribute file.

-sites

Specifies a comma-separated list of site names with **plm_report_extract** files exist in the **input_dir** location.

- If not specified, all sites files in the **input_dir** location are processed.
- (Optional) One site can be identified as the source site and one site as a the target site.

For example:

```
-sites=site1=src_site,site2,site3=target_site,site4
```

Sites other than the source site and the target site are treated as third sites.

Note:

Only one source site and one target site are allowed in the list of sites, and they cannot be the same site.

-report_dir

Specifies the directory to place output files.

-suggest_no_fixes

Suppresses listing suggestions for fixing the inconsistencies reported.

-verbose

Displays additional progress information.

-ods_extract_site

Specifies the ODS site name to include ODS information in the analysis.

-h

Displays help information.

EXAMPLES

- To run a consistency analysis on all extract files placed in a single folder:

```
plm_report_consistency_analysis -input_dir=d:\data
  -report_type=consistency_analysis -report_dir=d:\out
```

- To run a consistency analysis on a subset of the extract files placed in a single folder:

```
plm_report_consistency_analysis -input_dir=d:\data
  -report_type=consistency_analysis -report_dir=d:\out
  -sites=SiteOne,SiteTwo,SiteThree
```

- To produce a text version of binary files generated by the **plm_report_extract** utility:

```
plm_report_consistency_analysis -input_dir=d:\data  
-report_type=uncompress -report_dir=d:\out
```

plm_report_constraint_analysis

DESCRIPTION

Analyzes the target site, based on a source-site extract generated using **plm_report_extract**, and identifies conflicts in the key, database-level, unique indexes that can cause import issues at the target site.

For enhanced readability and further processing, the report generated by this utility can be imported into other applications such as Microsoft Excel.

The **plm_report_constraint_analysis** utility checks and reports potential violations on the following index-based constraints.

Index name	Class	Attributes
item_id_upper_index	Item	Item_id
item_revision_index	ItemRevision	items_tag,item_revision_id,sequence_id
ixr_index	ImanExportRecord	ixr_exported_object,ixr_target_site
itxr_index	ItemExportRecord	itxr_exported_object,itxr_target_site
par_index	PublicationAuditRecord	par_published_object,par_target_site
upper_lower_qual_index	AbsOccDataQualifier	qualifier_bvr,upper_qual_object,lower_qual_object
idincontext_index	Identifier	idfr_id,idcontext,suppl_context
list_name_index	EPMAssignmentList	list_name

The report contains a header in the following format to allow you to plan and track incremental site consolidation by reporting source and target sites and the files that are run:

```
Constraint Analysis Report
Constraint Analysis Report Created at [2013-02-14 22:37:15]
Source Site : tc101ms2
Target Site : tc101ms1
File(s) processed :
E:\tasks\extract\const_000001_tgt.xml
E:\tasks\extract\const_000002_tgt.xml
E:\tasks\extract\remote_object_001_tgt.xml
E:\tasks\extract\testcase_000001_tgt.xml
```

SYNTAX

```
plm_report_constraint_analysis -u=user-name {-p=password | -pf=file}
[-g=group] -input_dir=directory -report_dir=directory
-src_site=source-site-name [-verbose] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

input_dir

Specifies the directory location of input files generated by **plm_report_extract** utility.

report_dir

Specifies the directory location for the output report.

src_site

For constraint analysis, specifies the source site.

-verbose

Displays additional progress information.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

Either the **-item=***item-id* or the **-i=***input-file* argument must be supplied.

EXAMPLE

To perform constraint analysis on the target site for a given source site extract:

```
plm_report_constraint_analysis -u=tc-admin-user -p=password  
-g=group -input_dir=d:\data -report_dir=D:\out -src_site=Site311
```

purge_datasets

Removes (purges) old versions of datasets from the database and outputs a list of each dataset purged, along with the owning user and group.

Note:

Normally, Teamcenter stores a fixed number of dataset versions in the database. The maximum number of datasets retained is set using the **AE_dataset_default_keep_limit** preference.

Certain conditions prevent automatic purging of old datasets. For example, when a user does not have permission to purge a dataset owned by another user, or when a group is given read/write permission but not delete permission.

Additionally, datasets are not purged when the named references in **version0** do not match the named references in the latest dataset version. Use the **-skipInconsistencyCheck** argument to bypass this named references check. Use this argument only in situations in which you know why the named references differ and are confident that purging the older dataset versions will not result in loss of needed data. Possible situations include CAD integrations in which custom code is implemented, wrong coding exists from custom ITK programs, and PLM XML import.

Before using the **-skipInconsistencyCheck** argument, run this utility without the argument and review the output for any failed purges. Investigate all datasets that did not purge, correcting problems if necessary.

For example, **version0** of a dataset contains three named references: **a.txt**, **b.txt** and **c.txt**. The latest version of the same dataset contains two named references: **a.txt** and **b.txt**. Using this utility to purge this dataset fails because of the inconsistency between the named references. The utility logs an inconsistent data message.

In this situation, you should compare the named references between the versions and resolve the inconsistency if necessary.

- Compare the named references of the two versions in the rich client by choosing **View→Named References**.
- Correct the inconsistency by copying the **c.txt** named reference from **version0** to the clipboard, then pasting it into the latest version. The named references must be in the same order for both versions. Rerunning the utility would purge this dataset.

Alternatively, checking the dataset out, making changes, and checking it back in synchronizes the named references between the versions.

If it is not possible to synchronize the named references, for example, the datasets are already released (and thus read-only), then use the **-skipInconsistencyCheck** argument to purge the datasets.

SYNTAX

```

purge_datasets [-u=user-id {-p=password | -pf=password-file} -g=group]
[-b=beginning_anchor] [-e=ending_anchor] [-k=keep-limit]
[-set] [-report] [-replica_only] [-site=site-name]
[-start_date="DD-MMM-YYYY HH:MM:SS"]
[-end_date="DD-MMM-YYYY HH:MM:SS"]
[-itemidsfile=item-id-file-name]
[-itemKeyFile=item-key-file-name]
[-grmtypesfile=relation-types-file-name]
[-anchorfile=revision-anchor-file-name]
[-includeFolderContents]
[-skipInconsistencyCheck]
[-h]

```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, then the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-b

Specifies the first version anchor (*beginning_anchor*) of a contiguous series to be purged. The default value is **1**. Version anchors are objects that keep track of a set of versions of an object. Datasets are one such class of objects.

-e

Specifies the last version anchor (*ending_anchor*) of a contiguous series to be purged. The default value is **last**. Version anchors are objects that keep track of a set of versions of an object. Datasets are one such class of objects.

-k

Specifies the keep limit for purging.

For example, if the **-k** argument is set to **-k=4**, then a dataset is purged down to four versions.

If the **-k** argument is not used, then the current version limit of the dataset is used as the purge keep limit.

-set

For the datasets to be purged, sets the version limit to the **-k** argument value.

The **-set** argument has no effect if the **-k** argument is not used.

-report

Produces output showing the datasets to be purged, but does not purge the datasets from the database.

The output lists the item IDs of datasets that need to be purged. To use these IDs in the input file needed by the **-itemidsfile** option, manually copy the item IDs into the input file.

-replica_only

Specifies that only replica datasets are purged. The **-site** argument can be used in conjunction with the **-replica_only** argument to purge only the datasets replicated from a specific site.

-site

Specifies the site from which replica datasets are purged. Valid only in conjunction with the **-replica_only** argument.

-start_date

Specifies the start date/time for which datasets are purged.

-end_date

Specifies the end date/time for which datasets are purged. This argument is optional and must be used only with the **-start_date** argument.

-itemidsfile

Specifies an input file containing a list of item IDs. Use the **-itemidsfile** argument followed by the directory path and file name.

To obtain the item IDs, run the **purge_datasets** command with the **-report** argument.

In the input file, enter one item ID per line, or you can enter multiple item IDs on one line separated by commas.

For example, a file *item_uid.txt* might contain this list:

```
000041
000051
000078
```

-itemKeyFile

Specifies an input file containing a list of item keys <attr1=value1,attr2=value2,...>. Use the **-itemKeyFile** argument followed by the directory path and file name.

In the input file, enter one item key per line, or you can enter multiple item IDs on one line separated by commas.

-grmtypesfile

Specifies an input file containing a list of specified relation types to use. Enter one relation type per line, or enter multiple relation types on one line separated by commas.

If this parameter is not specified, then all relation types are used.

This parameter is valid only with the **-itemidsfile** argument.

-anchorfile

Specifies the name of a file that contains UIDs of revision anchors to process.

On very large databases, the number of objects processed may make this utility run slow or fail. Using an anchor file can prevent failures and increase the speed of the dataset purge process.

To generate a list of UIDs to process, use the **item_report** utility with its **-anchorfile** argument.

-includeFolderContents

Gets all the datasets related to the Item IDs specified for purge in the input file in **-itemidsfile**. This parameter is optional and is valid only with the **-itemidsfile** argument.

-skipInconsistencyCheck

Bypasses the consistency check of named references and purges previous versions of a dataset even if the named references in **version0** of the dataset are not the same as the named references in the latest version.

If **-skipInconsistencyCheck** is not specified, then a dataset is not purged if the named references in **version0** of the dataset are not the same as the named references in the latest version.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment..

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

The **purge_datasets** utility must be run by the a user with Teamcenter administration privileges. This automatically enables the bypass feature and purges old datasets regardless of owning user and group.

EXAMPLES

- Purge the range of contiguous datasets down to the specified keep limit, and set the version limit of the current dataset to the keep limit value:

```
$TC_BIN\purge_datasets -u=Tc-admin-user -p=password -g=group -b=beginning-  
anchor -e=ending-anchor -k=keep-limit -set
```

- Purge datasets with specified item ids, and also purge their related items:

```
$TC_BIN\purge_datasets -u=Tc-admin-user -p=password -g=group  
-itemidsfile=D:\workdir\item_uid.txt -includeFolderContents
```

review_volumes

Allows you to view volume file attributes regarding OS volumes (file size, last modification date, and so on) and to remove unreferenced operating system files from these volumes. This utility can generate a report file describing volume usage by various groups and users, as well as reporting any unreferenced operating system files, missing operating system files, and unreferenced Teamcenter files. Unreferenced operating system files can be deleted at the time a report file is generated or at a later time using a previously-generated report file as an input. The report file format is plain text (ASCII) and can be manually edited in order to not delete certain files. To prevent files from being deleted, remove any file names before using the report file as input. You can also save any deleted files to a ZIP format compressed file.

Note:

Before creating volumes, you must have an FMS server cache (FSC) installed and running, and you must set the **FMS_BootStrap_Urls** preference with the FSC host and port information.

The report file contains three main sections that list issues with Teamcenter files:

- Section 1: Unreferenced OS files

Lists files in the OS file system over a day old that are not referenced by Teamcenter. These are files that exist in the OS volume directory but are not referenced by any Teamcenter **TcFile** data objects. These are the OS files that are deleted when the **review_volumes** utility is run with the **-if** argument. During the **-if** run, these files are deleted only after a double-check that they are indeed not referenced.

- Section 2: Missing OS files

Lists files missing from the OS file that are referenced by Teamcenter. These are files that do not exist in the OS volume directory but are referenced by Teamcenter data objects. This situation may be remedied by copying the missing files from backed-up volumes or whole file caches. The **-if** run does not take any additional action on files reported in this section.

- Section 3: Unreferenced Teamcenter files

Lists files in the OS file system referenced from unreferenced **ImanFile** objects. These are files that exist as **TcFile** data objects, but those **TcFile** data objects are not referenced by any dataset data objects. For files that are listed in this section, the **-if** run deletes the **TcFile** object and the OS file from the volume. Teamcenter referential integrity ensures that the **TcFile** object deletions occur only if they are not referenced by other Teamcenter objects.

To illustrate the report output, assume that the Teamcenter database has the following contents:

Dataset	Reference type	Referenced TcFile	Volume file name of the referenced file
MyDataset1	-	-	-
MyDataset2	Text	MyFile.txt	eng_3219879864/ MYFILE209734EA_32470987.txt
MyDataset3	MSWordX	Schedule3.docx	eng_3219879864/ SCHE4867986324_3467.docx
	Zip	MaintenanceSchedules.zip	eng_3403297097f/ MAINTE49907r3732498_kjlewuih.zip
MyDataset4	File	4GMatrix.dat	eng_3403297097f/ 4GMAFJKSN39479679_3234777429869.dat
-	-	Lolapalooza.mid	eng_3219879864/ LOLAP214097409_32209874986.mid
-	-	VennDiagram.dia	eng_3219879864/ VENND329879864_379407.dia

Also assume that the volume contains the following files:

```
eng_3219879864/LOLAP214097409_32209874986.mid
eng_3219879864/MYFILE209734EA_32470987.txt
eng_3403297097f/MAINTE49907r3732498_kjlewuih.zip
eng_3403297097f/BARTFARGLE0321970_32986.tag
eng_3403297097f/FASTNR2139840397_3219473209.prt
```

The report lists the following:

- Unreferenced OS files

The following files exist in the volume but are not referenced by any Teamcenter data objects:

```
eng_3403297097f/BARTFARGLE0321970_32986.tag
eng_3403297097f/FASTNR2139840397_3219473209.prt
```

- Missing OS files

The following files do not exist in the volume but are referenced by Teamcenter data objects:

```
eng_3219879864/SCHE4867986324_3467.docx
eng_3403297097f/4GMAFJKSN39479679_3234777429869.dat
eng_3219879864/VENND329879864_379407.dia
```

- Unreferenced Teamcenter files

The following files exist as **TcFile** data objects, but these **TcFile** data objects are not referenced by any dataset data objects:

```
eng_3219879864/LOLAP214097409_32209874986.mid
eng_3219879864/VENND329879864_379407.dia
```

SYNTAX

```
review_volumes [-u=user-id {-p=password | -pf=password-file} -g=group]
-v=volume -rf= file-name | -if=file-name
[-of=file-name] [-zf=file-name] [ -lv]
[-parallel=number-of-parallel-processes] [-rfolder=folder-name]
[-noDeleteCheck]
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-v

Runs the utility against a single, specified volume. This argument is required unless the **-lv** or **-parallel** argument are specified.

-rf

Creates a report file listing unreferenced files in volumes.

Note:

The **-of**, **-if**, and **-zf** arguments do not work in combination with the **-rf** argument. When the **-rf** argument is present, only the report file generation is performed.

-if

Specifies the report file to be used as input to delete unreferenced files in volumes.

-of

Deletes and logs the results to a specified file. This argument must be supplied if the **-if** argument is used.

-zf

Saves deleted files to the specified **ZIP** file. The **.zip** extension is automatically appended to the file name if another extension is not specified.

-lv

Lists all volumes defined in the database.

-parallel

Specifies the number of volumes on which to simultaneously run this utility. You can use this argument to generate reports on all volumes defined in your database simultaneously (use the **-lv** argument to determine the number of volumes defined in your database). However, you must consider available computing resources while setting this value. Even if you have 800 volumes defined in your database, you might only have enough computing power to run five or ten processes in parallel.

This argument must be used with the **-rfolder** argument.

-rfolder

Specifies the folder in which the multiple reports generated by the **-parallel** argument are stored.

This argument must be used with the **-parallel** argument.

-noDeleteCheck

Turns off double-checking of OS files to be deleted. Double-checking ensures that OS files to be deleted are checked again for **ImanFile** references before deletion. This avoids accidental deletions when corrupt or tampered report files are used. Use this argument with the **-if** argument.

-h

Displays help for this utility.

ENVIRONMENT

As specified in Manually configure the Teamcenter environment..

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

In versions prior to 8.3.2, you could supply the **-rf** argument in combination with the **-of** argument in order to have the **review_volumes** utility simultaneously generate a report file and delete files from the specified volume.

Beginning with Teamcenter 8.3.2, this requires two separate steps:

1. Run the **review_volumes** utility with the **-rf** argument.
2. Use the **-if** and **-of** arguments to input the previously created **report** file and **delete volume** files.

This change in how arguments are processed was required by the need to improve scalability and performance of the **review_volumes** utility, for example, the addition of the new **-parallel** argument.

EXAMPLES

- To generate a report on a single volume, enter the following command on a single line:

```
review_volumes -u=user-id -p=password -g=group -v=volume -rf=file-name
```

- To delete files on a single volume from a previously executed report, enter the following command on a single line:

```
review_volumes -u=user-id -p=password -g=group -v=volume -if=file-name -of=file-name
```

- To list all volumes defined in the database, enter the following command on a single line:

```
review_volumes -u=user-id -p=password -g=group -lv
```

- To generate report on all volumes defined in the database, enter the following command on a single line:

```
review_volumes -u=user-id -p=password -g=group -parallel=number-of-parallel-processes  
-rfolder=folder-name
```

sitcons_accountability_chk

DESCRIPTION

Provides a list of nonconsolidated objects by a given class.

SYNTAX

```
sitcons_accountability_chk -u=user-name {-p=password | -pf=file} [-g=group]  
-report=file -class=class [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-report

Specifies the file, with absolute path, where the report is created. This report contains information about the unconsolidated objects.

-class

Specifies the class name of the object to check consolidation status. There is no default class. If the class is a workspace object, the object name and PUID are reported. Otherwise, only the PUID is reported.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

To generate the pending consolidation report by a given class:

```
sitcons_accountability_chk -u=tc-admin-user -p=password  
-g=group -report=D:\Temp\report.txt -class=Item
```

sitcons_extract_shared_vols

DESCRIPTION

From the source site, generates the volume map extract output used to generate the shadow volume at target site. This utility must be run at the source site. It extracts the list of volumes and generates the shared volume names based on the prefix and/or suffix given. This information is used to create and map shared volumes at the target site using the **sitcons_gen_shared_vols** utility.

SYNTAX

```
sitcons_extract_shared_vols -u=user-name {-p=password | -pf=file} [-g=group]
-outputfile=file -prefix=prefix_string -suffix=suffix_string -fsc_id= fsc-id
[-node_name=node-name] [-h]
```

You must provide either a **prefix** parameter or a **suffix** parameter, or both. When both are provided, both are used to derive the shared volume name.

The **prefix** parameter and the **suffix** parameter should include any delimiters required. The system does not add any delimiters between the parameters and the source volume name.

For example, with **-prefix=shadow** and **-suffix=_map**, the source volume **Test** has **shadowTest_map** for the shared volume name.

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-outputfile

Specifies the text file, with absolute path, into which the extract output is written.

-prefix

Specifies the prefix string added to the source volume name to derive the corresponding shared volume name.

-suffix

Specifies the suffix string added to the source volume name to derive the corresponding shared volume name.

-fsc_id

Specifies the FSC ID for the shared volume.

-node_name

Specifies the node name for the shared volume. If not provided, the system uses the name defined with the source volume.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

To extract source volume information from a source site which has:

```
sitcons_extract_shared_vols -u=tc-admin-user -p=password  
-g=group -prefix=map -fsc_id=FSC_hyi3w135_ntpriv_GMS135  
-node=hyi3w135
```

sitcons_fix_ixr

DESCRIPTION

Corrects export and publish records (IXR, ITXR, and PAR) at the third site and the source site after site consolidation object ownership transfer.

SYNTAX

```
sitcons_fix_ixr -u=user-name {-p=password | -pf=file} [-g=group]
{-f=fix_records | delete_records} [-inputfile=file-name]
[-target_site=site-name] [-source_site=site-name]
[-report=file-name] [-dryrun] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-f

Specifies the function that you want to perform. Valid values are **fix_records** and **delete_records**.

fix_records

When running **sitcons_fix_ixr** at a third site, fixes export and publish records (IXR, ITXR, and PAR) to reflect the target site as the owner of the objects that have been previously owned by the source site.

delete_records

When running **sitcons_fix_ixr** at a source site, deletes the export and publish records (IXR, ITXR, and PAR) of all applicable objects at source site whose ownership has been transferred to the target site.

-inputfile

Specifies the absolute path of the text file containing the item UIDs for which records need to be fixed.

-source_site

Specifies the source site name used when running the **sitcons_fix_ixr** utility at a third site to identify the objects that are owned by the source site.

-target_site

Specifies the site to which ownership of objects from the source site, previously owned by the source site, was transferred.

-report

Specifies the absolute path and file name of the report file. The report file contains fixes required for export and publish records (IXR, ITXR and PARs) at a third site and the source site.

If this argument is not specified, the report file is generated in the transient volume directory.

-dryrun

Generates the report file only. The candidate objects are searched and listed in the report. The report file contains fixes required for export and publish records (IXR, ITXR and PARs) at a third site and the source site.

If this argument is specified, the **-f** argument function is ignored. No deletions or updates are performed.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in Log files produced by Teamcenter.

RESTRICTIONS

With **-f=fix_records**, the **-target_site** and **{-inputfile | -source_site}** options are required.

With **-f=delete_records**, all other parameters are ignored.

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

- To fix records at a target site by specifying the item UID in the **inputfile** text file:

```
sitcons_fix_ixr -u=tc-admin-user -p=password -g=group -f=fix_records  
-target_site=SiteB -inputfile=input-file-path
```

- To fix records at a target site by specifying the source site only:

```
sitcons_fix_ixr -u=tc-admin-user -p=password -g=group -f=fix_records  
-source_site=site-name
```

- To delete records at a source site:

```
sitcons_fix_ixr -u=tc-admin-user -p=password -g=group -f=delete_records
```

sitcons_gen_shared_vols

DESCRIPTION

Creates the shared volumes at the target site based on a **sitcons_extract_shared_vols** output file containing a list of source shared volumes. The **sitcons_gen_shared_vols** also creates the **TIE_Volume_Map** preference and populates the preference values from the information in the input file.

Note:

This activity is essential for imported references from the source site to be mapped to the correct shared volumes by **tcxml_import** or **sitcons_replicate_mgr**.

If a site has a very large number of disk locations, set the **FSC_DelayedVolumeValidation** configuration parameter to **true**. This defers disk validation to a background thread, reducing the time the utility takes to create volumes.

You can improve the utility performance when generating a large number of volumes by increasing the memory setting for the FSC serving the volumes. The **FSC_MEM** environment variable sets the allocated memory for the FSC at 256 megabytes, by default during the FSC startup process. If the utility performance is unacceptable, Siemens Digital Industries Software recommends that you increase the value to 512M or 1024M and restart the FSC.

SYNTAX

```
sitcons_gen_shared_vols -u=user-name {-p=password | -pf=file} [-g=group]
-file=file [-override_mode=merge | override] [-stop_on_error] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-file

Specifies the input text file, with absolute path, containing one record for each shared volume to be created.

This input file is generated by executing the **sitcons_extract_shared_vols** utility on the source site with the required parameters. Each record contains required fields separated by commas following this syntax:

```
source_volume_name,shared_volume_name,shared_volume_node,fsc_id,shared_volume_path
```

-stop_on_error

Terminates the application if an error is encountered, such as, the input file does not contain the required number of fields in a record. The default is to log the error and continue running.

-override_mode

merge

If the **TIE_Volume_Map** preference exists and has values, the new values from the input file are added to the preference.

If the **TIE_Volume_Map** preference does not exist, it is created and populated.

override

If the **TIE_Volume_Map** preference exists and has values, the new values from the input file overwrite the existing preferences.

If the **TIE_Volume_Map** preference does not exist, it is created and populated.

This is the default mode.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

To create a shadow volume and **TIE_Volume_Map** preference:

```
sitcons_gen_shared_vols -u=tc-admin-user -p=password -g=group
  -file=/tmp/output_sitcons_extract_shared_vols/inpt.txt
  -override_mode=merge
```

The **sitcons_gen_shared_vols** utility is designed to be used only for site consolidation where *shared* volumes are being created.

- Shared volumes share a volume path with an existing volume definition.
- When shared volumes are created, the volume storage is shared by the target and the source installation.
- This utility is not intended for general creation of Teamcenter volumes.
- Do not use Teamcenter utilities to delete files in shared volumes. These files may appear to the site from which the utility is run to be unreferenced operating system files.

The general process to generate a shadow volume is as follows:

1. Execute the following command at the source site:

```
sitcons_extract_shared_vols -u=user -p=password -g=group name
  -outputfile=d:\volume_out.txt -suffix="_map" -fsc_id=FSC_ID
  -node=target_site_name
```

2. Copy the file created by the **sitcons_extract_shared_vols** utility to the target site.

3. Edit the volume path in the file to point to the path, such as the UNC path of the volume, which is accessible to the FSC at the target site.
4. At the target site, based on the output from the initial use of the **sitcons_extract_shared_vols** utility, execute the following command to create and populate the **TIE_Volume_Map** preference with the values in the input file:

```
sitcons_gen_shared_vols -u=user -p=password -g=group -file=d:\volume_out.txt
```

sitcons_gen_uidbatch

DESCRIPTION

Performs the function of generating a set of files containing a number of UIDs of a given class. These files can be used as input for replication for site consolidation purposes. Use the utility for replicating objects belonging to the following classes.

ImanEvent	VM_Policy	smlb0
ImanSubscription	HSM_Policy	smlb1
PDI_reprev_cache	Migration_Distribution	smllabel
PLMAppUID	Migration_Pending	stxt
GSIdentity	Migration_Report	sysd
Envelope	bldb0	
CCObject	bldb1	
StructureContext	unct_dict	
EPMJob	icml	

Additionally, you can use this utility for all Service Lifecycle Management (SLM) classes that are not decedents of the **Item** class and Pro/ENGINEER family classes and the associated generic assembly objects that must be included to transfer the entire Pro/ENGINEER family to the target site.

SYNTAX

```
sitcons_gen_uidbatch -u=user-name {-p=password | -pf=file-name}
[-g=group-name] { {-class=class-name | -genProEInstance}-outdir=folder}
[-batch_size=batch-size] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-class

Specifies the name of the class to extract and write the instance UIDs to text files. This argument is required if the **-genProEInstance** argument is not specified.

Note:

If you set the **-class** argument value to **Folder**, the **Home**, **Mailbox**, and **Newstuff** folders are skipped by the utility because they are not intended for site consolidation activity.

-genProEInstance

Generates a list of item IDs of Pro/ENGINEER family instances and generic assemblies that must be included as primary objects during export to successfully transfer the entire Pro/ENGINEER family to the target site. This argument is required if the **-class** argument is not specified.

-outdir

Specifies the output folder (with absolute path) where the files containing UIDs are generated. This argument is required.

-batch_size

Specifies the number of UIDs that each file can contain. If this argument is omitted or is invalid, the **1000000** default value is used.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

EXAMPLES

To generate files containing 5000 UIDs for **ItemEvent** class objects:

```
sitcons_gen_uidbatch -u=tc-admin-user -p=password -g=group -class=ItemEvent  
-outdir=/tmp/datatransfer/ItemEventUIDs.txt -batch_size=5000
```

sitcons_replicate_mgr

DESCRIPTION

Replicates the data from a source site to the target site using the TC XML low-level transfer options for site consolidation. This utility requires Teamcenter Integration Framework to be running and configured for site consolidation.

SYNTAX

```
sitcons_replicate_mgr -u=user-name {-p=password | -pf=file} [-g=group]
{-sync | {-inputfile=file-name | -folder=folder-name | -inputuidfile=file }}
[-sitename=site-name] [-optionset=option-set] [-chunksize=size] [-reason=reason]
[-immediate=True | False] [-notify=True | False] [-emailaddr=email_address,email_address] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-sync

This switch is used for synchronization. This is based on the source site accountability tables and not input files or folders.

-inputfile

Specifies the text file, with the absolute path, containing the item IDs of source site items. This file format is supported by **tcxml_export -low_level**.

The input file contents must use the following format:

```
item_id=export_001
item_id=M2Item1_001,object_name=M2Item_name1,object_type=M2Item1
```

-folder

Specifies the name of an existing folder at the source site from which contents are to be exported from the source site and imported into the target site.

-inputuidfile

Specifies the file name and absolute path to text file containing the list of UIDs of objects to be exported from the source site and imported into the target site.

The input file contents must use the following format:

```
wUTBAKxKAAgcRA
wURBAKxKAAgcRA:Fnd0GeneralAudit
```

-sitename

Specifies the name of the target site into which the information exported from the source site is to be imported.

-optionset

Specifies the name of the option set to be used for traversal. The default option set is **SiteConsolidationDefault**.

-chunksize

Specifies the number of objects in each import set of objects for multiple parallel imports.

-reason

Specifies the reason for export. If blank characters are embedded, the entire string must be inside quotation marks.

-immediate

Directs Teamcenter Integration Framework to schedule the transaction immediately.

True

Starts the export process immediately.

False

Starts the process at the preconfigured off-hours time. This is the default.

-notify

Specify whether preconfigured Teamcenter Integration Framework users are notified, and whether messages are sent to the email addresses listed in the **-emailaddrs** control argument.

True

Sends email.

False

Does not send email. This is the default.

-emailaddrs

Lists email addresses, comma delimited, for notification of ownership transfer, in addition to users configured by Teamcenter Integration Framework

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

- To replicate from one site to another by item ID:

```
sitcons_replicate_mgr -u=tc-admin-user -p=password -g=group
-immediate=True
-sitename=hyi3w165 -inputfile=/tmp/datatransfer/inputdata.txt
```

- To replicate from one site to another by UID:

```
sitcons_replicate_mgr -u=tc-admin-user -p=password -g=group  
-immediate=True  
-sitename=hyi3w165 -inputuidfile=/tmp/datatransfer/inputdata.txt
```

- To synchronize data:

```
sitcons_replicate_mgr -u=tc-admin-user -p=password -g=group  
-immediate=True  
-sitename=hyi3w135 -sync
```

sitcons_user_folders

DESCRIPTION

Extracts the contents of the users' **Home**, **Newstuff**, and **Mailbox** folders from the source site and updates the target site.

SYNTAX

```
sitcons_user_folders -u=user-name {-p=password | -pf=file} [-g=group]
{-mode=extract | update} -inputfile=file-name [-report=file-name] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-mode**extract**

At the source site, extracts the contents of the users' **Home**, **Newstuff**, and **Mailbox** folders.

update

At the target site, updates user folders contents.

-inputfile

Specifies the input file with absolute path.

- When **sitcons_user_folders** is run at the source site with the **-mode=update** argument, contains folder content details. The **-inputfile** argument should use the file specified by the **-report** argument when **sitcons_user_folders** was run at the source site in the extract mode. This argument is required in **update** mode.
- When **sitcons_user_folders** is run in **extract** mode, contains user IDs. If this argument is not provided when run in **extract** mode, information is extracted for all users.

-report

When **sitcons_user_folders** is run at the source site in the **extract** mode, this argument specifies the name, with absolute path, of the extract output file.

When **sitcons_user_folders** is run at the at the target site with **-mode=update**, the **-inputfile** argument uses the file generated at the source site by the **extract** mode.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

- To extract information from the source site:

```
sitcons_user_folders -u=tc-admin-user -p=password -g=group -mode=extract  
-report=/tmp/reports/siteA_user_folders.txt
```

- To extract the information from the source site for given input:

```
sitcons_user_folders -u=tc-admin-user -p=password -g=group -mode=extract  
-inputfile=/tmp/userdetails/userids.txt  
-report=/tmp/reports/siteA_user_folders.txt
```

- To update the folders, run the utility at the target site:

```
sitcons_user_folders -u=tc-admin-user -p=password -g=group -mode=update  
-inputfile=/tmp/reports/siteA_user_folders.txt
```

sitcons_xfer_owner_mgr

DESCRIPTION

Orchestrates transfers of object with ownership from a source site to the target site during site consolidation. The ownership change takes place at the target site first followed by the change at the source site. Ownership change is performed only for objects replicated by the site consolidation tools. Preview options allow you to display:

- Source site objects that have been replicated at the target site by the site consolidation tool.
- Source site objects that have been replicated at the target site and are inconsistent with the source site objects.
- Source site objects that have been replicated at the target site by Multi-Site Collaboration and, therefore cannot have their ownership changed by this utility.

This allows you to determine the effect of running the tool at a specific site without actually changing the ownership of any objects.

Note:

For performance reasons, the site consolidation utilities used to change ownership of objects use SQL calls, not the POM API. Rich client refresh works only when POM calls are used to change an attribute, including ownership changes. Therefore, ownership changes that the utilities enact during an active rich client session are not reflected in the interface. You must restart all active rich clients to get the changes. This is normally not an issue because user access must be prohibited during the time critical period when this utility is used.

SYNTAX

```
sitcons_xfer_owner_mgr -u=user-name {-p=password | -pf=file} [-g=group]
{[-change_ownership_to=target-site-id] | [-dryrun [-startDate=MM/DD/YYYY
-endDate=MM/DD/YYYY ] [-source_extinct] ] }
[-notify={True | False } [-emailaddrs= address1, address2, ... ] ] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you

do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager (TEM) prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-change_ownership_to

Specifies the site ID of the target site. This argument is required when the utility is run at the source site. If this argument is omitted, the utility uses the local site as the target site. If the **-dryrun** argument is specified, the ownership transfer is not performed.

-dryrun

Generates a report to preview the ownership change of objects transferred by the site consolidation tools, and identifies inconsistencies.

To preview the objects at the target site that are owned by the source site and are inconsistent with the source site due to modifications made to them at the target site post-replication, set the **TIE_DRYRUN_VALIDATION=TRUE** preference at the target site.

-startDate

Specifies the starting date for replicas and inconsistencies reported by the **-dryrun** argument. This argument is used in conjunction with the **-endDate** argument and is ignored if the **-dryrun** argument is not specified. The date supplied must be in supplied in a *MM/DD/YYYY* format.

-endDate

Specifies the ending date for replicas and inconsistencies reported by the **-dryrun** argument. This argument is used in conjunction with the **-startDate** argument and is ignored if the **-dryrun** argument is not specified. The date supplied must be in supplied in a *MM/DD/YYYY* format.

-source_extinct

Generates the list of replica items at the target site which are owned by the source site and which were not transferred by site consolidation tools.

This argument can be used only when the **-dryrun** argument is used and is ignored otherwise.

-notify

Specifies whether the users that are configured to receive notification of ownership change in Teamcenter Integration Framework are notified through email of ownership changes performed by this utility. If you specify **True**, the users are notified. If you include the **-emailaddrs** argument, the email addresses identified by that argument are also notified. If this argument is not specified, no notifications are sent.

-emailaddrs

Specifies a comma delimited list of email addresses of recipients that receive notification of ownership changes performed by this utility.

-h

Displays help information.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

The **SITCONS_AUTH_KEY** environment variable must be set to run this utility. Open a support case on Support Center to get this license key.

EXAMPLES

To transfer ownership from one site to another:

```
sitcons_xfer_owner_mgr -u=tc-admin-user -p=password -g=group  
-change_ownership_to=528122928
```

tcxml_confirm_export

DESCRIPTION

For data replicated at target site, updates the status at the source site.

SYNTAX

```
tcpxml_confirm_export -u=user-name {-p=password | -pf=file} [-g=group]  
[-file=import_result_file] [-low_level] [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges. If this argument is used without a value, the operating system user name is used.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

If used without a value, the system assumes a null value. If this argument is not used, the system assumes the *user-id* value to be the password.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-low_level

Specifies the fast low level export mode. This is required for site consolidation.

Using the **-low_level** argument requires the **SITCONS_AUTH_KEY** environment variable be set to a valid license key. Open a support case on Support Center to get this license key.

-file

Specifies the location of the *filename_import_result.txt* file generated during import. The value can be either an absolute path (full path name) or a relative path name.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

EXAMPLES

- To confirm data import:

```
tcxml_confirm_export -u=tc-admin-user -p=password -g=group  
-low_level -file=filename_import_results.txt
```

tcxml_export

Exports objects from Teamcenter in the TC XML format. If there are files for export, the utility creates a File Management System (FMS) read file ticket and saves it in the output XML file for each file.

Note:

Do not use the **tcxml_export** utility for exporting administration data. For exporting administration data, use the **admin_data_export** utility.

SYNTAX

```
tcxml_export [-u=user-id [-p=password | -pf=password-file] [-g=group] ]
-file=output-xml-file {-item=item-id [-rev=revision-id] | -folder=folder-name |
-class=POM-classname | -uid=uid-of-object |
-item_key=attr-name1=value, attr-name2=value, ... |
-inputfile=file-name | -inputuidfile=file-name}
[-transfermode=transfer-mode-name | -optionset=transfer-option-set-name]
[-targetsites=list-of-target-site-ids]
[-transferownership]
[-sync]
[-incrementalChangeDelta]
[-force_reexport]
[-reason=reason-for-export]
[-revrule =revision-rule]
[-bomlevel =desired-bom-level]
[-svrule=saved-variant-rule-name [:rev-id:rule-id]]
[-processUnconfiguredByOccEff]
[-processSuppressedOcc]
[-processUnconfiguredVariants]
[-processUnconfiguredChanges]
[-baseline_id]
[-baseline_rev]
[-generateBOMIndex]
[-fromBOMIndex]
[-xsl=xsl-file-name]
[-session_options=option-1:value-1,option-2:value-2,...option-n:value-n]
[-session_options_file=options-file-name]
[-requiredLang=locale-code-1, locale-code-2, ..., locale-code-n]
[-allowedLang=locale-code-1, locale-code-2, ..., locale-code-n]
[-briefcase]
[-dryrun [=validateXMLBeforeXslt]]
[-validate [=validateXMLBeforeXslt]]
[-low-level {-inputfile=file-with-item-ids | -inputuidfile=file-with-uids}
[-bulk_extract]
[-input_criteria=class-name{attribute1=value1,attribute2=value2,...attributen=valuen}]
[-force_retraverse]]
```

[-verbose]
[-h]

ARGUMENTS

-u

Specifies the user ID.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-file

Specifies the output file name. The value can be either an absolute path (full path name) or a relative path name. By default, the output file contains XML. When the **-dryrun** or **-validate** option is specified, the file contains an HTML report of any warnings or errors in the TC XML. When using the **-dryrun** or **-validate** option, append a **.html** extension to the file name.

-item

Specifies the **item_id** values of one or more items to be exported. Multiple values are separated by colons.

-rev

Specifies the revision ID of the item to be exported. If this argument is not specified, the configured revision (either specified or default) is exported for the item.

-folder

Specifies the folder containing objects to be exported.

-class

Specifies a class name. Instances of this POM class are exported. The following workspace object names are valid:

- **Item**
- **ItemRevision**
- **Folder**
- **Dataset**
- **Alias**
- **ImanFile**
- **ImanRelation**
- **ReleaseStatus**
- **IdContext**
- **Identifier**
- **PSBOMView**
- **PSBOMViewRevision**
- **TransferMode**
- **TransferOptionSet**

-uid

Specifies the UID of an object (one object only).

-item_key

Specifies a comma-delimited list of attributes used to identify the object to be exported.

-item_key supports multifield keys using the following form:

```
-item_key=attribute_name_1=value_1, attribute_name_2=value_2, ...
```

-inputfile

Specifies the name of a text file containing the item IDs of objects you want to export. **-inputfile** also supports multifield keys. The file must have separate lines for the item attributes you want to use to identify the objects, for example:

```
item_id=export_001
item_id=M2Item1_001,object_name=M2Item_name1,object_type=M2Item1
```

-inputuidfile

Specifies the name of a text file containing the UIDs of objects you want to export. The file must have separate lines containing the object UID. For lightweight object UIDs, provide the UID and the object class separated by a colon.

-transfermode

Specifies the transfer mode name used to export the objects. If this argument is not specified, the utility uses a default transfer mode. See restriction [2](#).

-optionset

Specifies the transfer option set name used to export the objects. Mutually exclusive with the **-transfermode** argument. If you specify both, the command does not fail. However, the transfer

mode indicated by this **-optionset** argument takes precedence over the transfer mode specified by the **-transfermode** argument. See restriction 3.

-targetsites

Specifies a comma-delimited list of destination site IDs. If used with the **-transferownership** argument, **-targetsites** must contain only one site ID.

-transferownership

Indicates that this export transfers the ownership of exported objects to the specified target site. You must specify only one site ID in the **-targetsites** argument if you use this argument.

-sync

Indicates that this export is for data synchronization.

-incrementalChangeDelta

Exports modified objects tracked by configured incremental change as a partial structure export.

Because incremental change data is configured data, you must specify a configured transfer option set for the **-optionset** argument. The **-processUnconfiguredChanges** and **-processSuppressedOcc** arguments are ignored if they are included.

-force_reexport

Repeats the most recent configured export specified by the values of the following arguments: **-item**, **-targetsites**, **-revrule**, and **-svrule**.

Configured exports of high level and low level TC XML to managed or unmanaged sites are supported. Full and partial BOM exports are also supported.

Objects do not need to be marked for ownership transfer. The same ownership transfer specifications are used as in the previous export. Objects with their ownership transferred in the previous export are also included in this export.

If the previous export meeting this criteria was a full export, the full export is repeated, including changes since the previous full export. If there is no previous export, a full export is created.

-reason

Specifies the reason for this export.

-revrule

Specifies the revision rule used to configure the exported BOM with the specified item as the top line.

-bomlevel

Specifies the level in the BOM.

-svrule

Specifies a saved variant rule to use when configuring the exported BOM. Optionally, specify a particular revision of a variant rule object by providing its revision ID and rule ID separated by

colons. Variant rule revision use is supported for only **Cfg0VariantCriteria**, a revisable subclass of the **VariantRule** object.

-processUnconfiguredByOccEff

Exports **BOMLine** objects that are not configured for occurrence effectivity.

-processSuppressedOcc

Exports suppressed **BOMLine** objects.

-processUnconfiguredVariants

Exports **BOMLine** objects that are not selected by **BOMLine** object's variant conditions.

-processUnconfiguredChanges

Exports **BOMLine** objects configured out of the BOM by incremental change.

-baseline_id

Specifies the baseline ID for exporting a configured minor revision.

-baseline_rev

Specifies the baseline revision ID for exporting a configured minor revision.

-generateBOMIndex

Saves **BOMLine** data to persistent tables. The TC XML data is not serialized.

-fromBOMIndex

Exports the **BOMLine** data directly from the persistent cache without configuring and expanding the BOM.

-xsl

Specifies the output XSL file to apply to the TC XML file after export.

-session_options

Specifies a comma-separated list of option name-value pairs. Names and values are delimited by colons, for example:

```
ContinueOnError:TRUE,GenerateReport:TRUE
```

-session_options_file

Specifies an ASCII file containing session option settings using option name-value pairs. Use this option instead of the **-session_options** option when you are reusing long lists of session name-value pairs and when values contain system-reserved special characters.

Place one name-value pair on each line of the file as follows:

```
option_1:value_1
option_2:value_2
.
.
```

```
option_n:value_n
```

Set an option only once in the file.

-requiredLang

Specifies a list of comma separated locale values. This list is used to ensure that localized attributes in the exported data have at least one representation that can be used as the attribute primary language at the importing site. It also defines a priority order for the exporter to determine the attribute primary language. The valid locale values must match the Java locale naming convention that consists of two groups of two-character identifiers separated by an underscore character (`_`) for a particular combination of language and region. For example, **zh_CN** represents Simplified Chinese in China and **en_US** represents English in the United States.

-allowedLang

Specifies a list of comma separated locale values. This list is used to get additional representations for localized attributes in the exported data for use at the importing site. The valid locale values must match the Java locale naming convention that consists of two groups of two-character identifiers separated by an underscore character (`_`) for a particular combination of language and region. For example, **zh_CN** represents Simplified Chinese in China and **en_US** represents English in the United States.

-briefcase

Specifies the output be formatted as a briefcase file. The file is specified with the **-file** option. **-briefcase** is only supported when exporting to managed sites. Sites must be specified using **-targetsites**.

-dryrun

Specifies a simulated low-level TC XML export be run and the exported TC XML be validated. Warnings and errors in the TC XML are reported in the file specified by **-file**. If **-dryrun=validateXMLBeforeXslt** is specified, the internal (pre-export) TC XML is validated instead of the exported TC XML. A briefcase file not created in either case. **-dryrun** cannot be used with the **-validate** option.

-validate

Specifies that the exported TC XML be validated. If no warnings or errors are detected in the TC XML, the briefcase file is created. If warnings or errors are detected in the TC XML, they are reported in the file specified by **-file** (with a `.html` extension) and no briefcase file is created. If **-validate=validateXMLBeforeXslt** is specified, the internal (pre-export) TC XML is validated instead of the exported TC XML. **-validate** cannot be used with the **-dryrun** option.

-h

Displays help for this utility.

FAST EXPORT ARGUMENTS

The following arguments support low-level fast export functions used for site consolidation activities. The use of these arguments requires the **SITCONS_AUTH_KEY** environment variable be set to a valid license key. Open a support case on Support Center to get this license key.

Export files created using the low-level export do not contain global stable identifier (**GSID**) attributes.

-low_level

Performs fast export using POM-level APIs as a DBA user. You must specify this argument to use any of the fast export arguments.

In addition to the other fast export arguments, you can use any of the following standard **tcxml_export** arguments when you specify the **-low_level** argument.

- folder
- file
- targetsites
- transfermode
- optionset
- reason

Note:

The **-requiredLang**, **-allowedLang**, and **-transferownership** arguments are ignored if you supply them with the **-low_level** argument.

-bulk_extract

Extracts product data into a briefcase (.bcz) file that contains low-level TC XML and associated physical files used to bulk load the data into a test environment. This briefcase file is explicitly for a test environment and cannot be used for exchanging data with suppliers.

You can supply the **-optionset** argument if the transfer options set has its **opt_bulk_extract_bcz** option set to **TRUE**. If you do not specify the **-optionset** argument, the utility uses the **UnconfiguredBulkExtractDefault** transfer option set.

The **-force_retraverse** argument is set with the **-bulk_extract** argument. The following arguments are ignored if you supply them with the **-bulk_extract** argument:

- allowedLang
- incrementalChangeDelta
- requiredLang
- sync
- targetsites
- transferownership

If you do not have read and import privileges on an object when using the **-bulk_extract** argument, the object is exported as a stub in the TC XML file and the object's status in the export log is listed as `STUB_INSUFFICIENT_PRIVILEGE`. For example:

```
id14 [wuQtjENZAAgcRA] of type [Item ] - STUB_INSUFFICIENT_PRIVILEGE
```

-input_criteria

Identifies the criteria for specifying root objects for export. The class name and attribute values are used to search for the object or objects you want exported. Only single value attributes, including attributes from the parent classes, are supported. Subclasses are not included. Only the AND condition is allowed between different attributes. You can specify an attribute only once. Wildcard characters are supported as defined in the **TC_pattern_match_style** preference.

For example, to export all items with a 6 character item ID starting with **0000** and the object name starting with **Top**:

```
tcxml_export -input_criteria=Item{item_id=0000??,object_Name=Top*}
-bulk_extract -file=abc_top.bcz
```

No other special character operators are supported. For example, the following characters are not supported and cannot be used in the class name, attribute name, or attribute value:

```
{ } = ,
```

The attribute value for a date range must be in the following format:

```
attribute-name="start-date to end-date"
```

For example, to specify objects created from 20 March to 1 April:

```
creation_date="20-Mar-2014 04:00 to 1-Apr-2014 04:00"
```

For an object reference type attribute, use the UID as the attribute value. This is true for any TC XML export.

-inputfile

Specifies a file that contains a list of item IDs indicating items to export using fast low-level export. You must include either this argument or the **-inputuidfile** argument when using the **-low_level** argument.

If your Teamcenter environment uses multifold key identifiers, you must specify the multifold key values for the **item_id** attribute as a list of comma-separated values in the input file, for example:

```
item_id=M2Item1_001,object_name=M2_Item_name1,object_type=M2Item1
```

The input file may contain both multifold key and standard item ID values, for example:

```
Item_id=Ace1
Item_id=lor1,object_name=fixedPl,object_type=type
Item_id=lor2,object_descr=acmetool,object_type=type
```

-inputuidfile

Specifies a file that contains a list of UIDs indicating items to export using fast low-level export. You must include either this argument or the **-inputfile** argument when using the **-low_level** argument.

-force_retraverse

Forces retraversal of previously replicated or exported objects during fast low-level export.

-verbose

Displays additional details during progress updates.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

1. Not all PLM data is supported. For a list of objects that are supported, see *Teamcenter Data Exchange*.
2. If you specify the **TransferMode** or **TransferOptionSet** object as the **-class** argument value, you are not required to specify the **-transfermode** or **-optionset** arguments. A predefined transfer mode is used for exporting these objects. If these arguments are specified they are ignored.
3. To export related dataset files, you must specify the **-transfermode** argument. The arguments value must be set to an option set containing closure rules that traverse dataset files related to the primary object. The **TIEExportDefaultTM** transfer mode contains a standard option set that can be used for this purpose.

EXAMPLES

- Select an item and export the item and its attachments using the default export transfer mode. The output XML file, **exportitem.xml**, is created in the directory where this command is executed.

```
tcxml_export -item=item_ID -file=exportitem.xml
```

- Select an item revision and export its attachments using the default export transfer mode. The output XML file, **itemrev.xml**, is created in the directory where this command is executed.

```
tcxml_export -item=item_ID -rev=item_rev -file=itemrev.xml
```

- Export the contents of the **exportObjects** folder using the default export transfer mode. If objects in the folder are supported objects, they are also exported. The output XML file, **folder.xml**, is created in the directory where this command is executed.

```
tcxml_export -folder=exportObjects -file=folder.xml
```

- Export item **000001** using the **TIEUnconfiguredExportDefault** transfer mode.

```
tcxml_export -item=000001 -file=exportitem.xml  
-optionset=TIEUnconfiguredExportDefault
```

- Synchronize item **000001**.

```
tcxml_export -item=000001 -file=itemsync.xml  
-optionset=TransferOptionSet -sync -reason=ItemIsOutDated
```

- Export the **Latest Working** revision of the **Top1** item using the **TIEConfiguredExportDefault** transfer option set.

```
tcxml_export -item=Top1 -optionset=TIEConfiguredExportDefault  
-revrule="Latest Working" -file=D:\temp\Top1_HL.xml
```

- Export a partial structure that includes only the changes to the **Top1** assembly that are tracked by configured incremental change:

```
tcxml_export -item=Top1 -targetsites=-2054508072  
-optionset=TIEConfiguredExportDefault  
-revrule="Latest Working" -sync -incrementalChangeDelta -file=/tmp/  
0001_delta.xml
```

- Fast export the objects identified in the **inp.txt** file using the **VARIANTRULE1** saved variant rule to site **56781234**.

```
tcxml_export -low_level -inputfile=d:\Temp\inp.txt  
-file=d:\Temp\top1_ll.xml  
-optionset=TIEConfiguredExportDefault -svrule=VARIANTRULE1  
-targetsites=-2054508072
```

- Fast export the **Latest Working** revisions of the objects, including suppressed BOM lines and BOM lines configured out by incremental changes.

```
tcxml_export -low_level -inputfile=d:\Temp\inp.txt  
-file=d:\Temp\top1_ll.xml -optionset=TIEConfiguredExportDefault
```

```
-revrule="Latest Working" -svrule=VARIANTRULE1 -processSuppressedOcc
-processIncrementalChanges
```

- Simulate a low-level briefcase export, validating the exported low-level TC XML and generating an error report (and no briefcase file).

```
tcxml_export -file=d:\Temp\report.html -item=root_obj_ID
-optionset=TIEConfiguredLLBCZExportDefault -briefcase
-targetsites=site_ID -dryrun
```

- Simulate a low-level briefcase export, validating the internal (pre-export) low-level TC XML and generating an error report (and no briefcase file).

```
tcxml_export -file=d:\Temp\report.html -item=root_obj_ID
-optionset=TIEConfiguredLLBCZExportDefault -briefcase
-targetsites=site_ID -dryrun=validateXMLBeforeXslt
```

- Perform a low-level briefcase export, validating the exported TC XML and creating a briefcase file if no warnings or errors are encountered. If errors or warnings are encountered, create a report file.

```
tcxml_export -file=d:\Temp\briefcase.bcz -item=root_obj_ID
-optionset=TIEConfiguredLLBCZExportDefault -briefcase
-targetsites=site_ID -validate
```

The following are site consolidation examples:

- To export the objects to a specified site.

```
tcxml_export -optionset=SiteConsolidationDefault
-targetsite=-2054508072 -inputfile=d:\input.txt
-file=d:\out.xml -low_level
```

- To synchronize the objects already exported (low level) to a specified site.

```
tcxml_export -sync -file=d:\out.xml -low_level
```

Special cases for **tcxml_export**:

- Export in-process workflow Items:
 1. Export the item in the workflow.
 2. Search for workflow job corresponding to that item.
 3. Copy this job and paste it into a new folder using a unique name.

4. Export this folder to export the job.

When a workflow process is exported with transfer ownership to the target site, the **My Worklist→Inbox→Tasks to Perform** folder at the target site does not display the task. To display the task, the affected users must delete the **Inbox** and restart their client.

Similarly, to display tasks in **ResourcePool Inbox**, delete its **Tasks to Track** and **Tasks to Perform** subfolders.

- Export a collaboration context object (CCO):
 1. In the rich client, search for the CCO.
 2. Copy the CCO and paste it into a new folder with a unique name.
 3. Export this folder to export the CCO and the entire structure context.
 4. Export the associated product structures.
- Low-level synchronization process example:
 1. Create the tables and triggers using an SQL script before exporting the data.

You can use an edited version of the following sample script, located in your **TC_ROOT/install/sitecons** directory, to manage the tables in separate tables spaces for the site consolidation tables. Before you run the script, edit the highlighted parameter values for your site.

```

sitcons_create_tablespace.sql
/* Copyright Siemens Product Lifecycle Management Software Inc.
   All Rights Reserved. */
/* This is a sample script that can be used by the administrator.
   The following parameters namely datafile, size, autoextend on maxsize,
   extent management local uniform size are the variables that need to be
   changed as required. Also, the tablespace can be named as desired */
/* Creates a separate table space to be used later for ACCT_TABLE creation */
create tablespace TCSITCONS datafile
  'D:\oracle\product\10.2.0\oradata\test\TCSITCONS.dbf'
size 10M
autoextend on maxsize 100M
extent management local uniform size 64K;
/* Creates the table ACCT_TABLE using above table space */
create table ACCT_TABLE(
exp_obj_uid varchar2(15) PRIMARY KEY,
led date,
island_anchor_uid varchar2(15),
state NUMBER)
tablespace TCSITCONS;
create table SCRATCH_TABLE(
puid varchar2(15),
lsd date,
trigger_condition NUMBER)
tablespace TCSITCONS;

```

```

CREATE INDEX "SCRATCH_TABLE_INDEX" ON "SCRATCH_TABLE" (puid);
create or replace trigger fast_sync_add_trigger
before insert on PPOM_OBJECT
referencing new as newRow
for each row
BEGIN
insert into scratch_table values (:newRow.puid, :newRow.plsd, '8');
END;
/
create or replace trigger fast_sync_delete_trigger
after delete on PPOM_OBJECT
referencing old as oldRow
for each row
BEGIN
insert into scratch_table values (:oldRow.puid, :oldRow.plsd, '9');
END;
/

```

- Export the data using the **tcxml_export** utility in low-level mode:

```

tcxml_export -optionset=SiteConsolidationDefault
-targetsite=-2054508072 -inputfile=d:\input.txt -file=d:\out.xml -low_level

```

- Import the data at the target site using the **tcxml_import** utility in low-level mode:

```

tcxml_import -file=d:\out.xml -low_level

```

- Copy the import-generated **out_import_results.txt** file to the source site and run the **tcxml_confirm_export** utility in low-level mode:

```

tcxml_confirm_export -file=d:\out_import_results.txt -low_level

```

- Add, delete, and change the low-level exported data as needed.

- Run the **tcxml_export** utility with the **-sync** argument.

```

tcxml_export -file=d:\sync.xml -low_level -sync
-optionset=SiteConsolidationDefault -inputfile=d:\input.txt

```

- Import the generated XML file using **tcxml_import** low-level mode.

- Repeat the earlier step to copy the import-generated **out_import_results.txt** file to the source site and run the **tcxml_confirm_export** utility in the low-level mode.

```

tcxml_confirm_export -file=d:\out_import_results.txt -low_level

```

Fast synchronization operations depend on the time stamp on the local server machine where an object is saved and edited. Therefore, modifying objects on different machines with different system times may influence the identification of out-of-sync status.

tcxml_import

Imports objects into Teamcenter from a TC XML file. The **tcxml_import** utility uses the **PS_bvr_import_new_occ** operation.

The low-level TC XML arguments of this utility provide access to site consolidation functionality. To use those arguments, you must set a license key value for the **SITCONS_AUTH_KEY** environment variable. Open a support case on Support Center to get this license key.

Caution:

Do not use the **tcxml_import** utility for importing administration data. For importing administration data, use the **admin_data_import** utility.

Property values can be updated in bulk. See *Process for updating property values in bulk*.

SYNTAX

```
tcxml_import
[-u=user-id{-p=password | -pf=password-file} -g=group]
-file=xml-file-name
[ [-xsl=xsl-file-name | -mappingcontrolfile=map-file-name]
[-errorcontinue={yes | no} ]
[ [-site=site-name]
[-transfermode=transfer-mode]
[-optionset=option-set-name] ] |
[ [-scope_rules [-scope_rules_mode=ignore | overwrite] ] ]
[-requiredLang=locale-code-1, locale-code-2, ..., locale-code-n]
[-allowedLang=locale-code-1, locale-code-2, ..., locale-code-n] ] |
[-low_level {-file=file-name}
[-bulk_load {-site=site-name -file=file-name} ]
[-tcfile_import]
[-islands=island-list]
[-bypassSiteCheck]
[-session_options=option-1:value-1, option-2:value-2,...option-n:value-n]
[-session_options_file=options-file-name]
[-briefcase]
[-dryrun [=validateXMLBeforeXslt]]
[-validate [=validateXMLBeforeXslt]]s
[-h]
```

ARGUMENTS

-u

Specifies the user ID.

If Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO rather than being authenticated against the Teamcenter database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-file

Specifies the input TC XML file containing the objects to import into Teamcenter.

-xsl

Specifies the input XSL file to apply to the TC XML file before import.

-mappingcontrolfile

Specifies the mapping control file to apply to the TC XML file before import.

-errorcontinue

Indicates whether to continue the import after encountering an error. The default value is **yes**. If you specify **no** and the utility encounters an error, the utility rolls back all of the changes performed during the current import.

-site

Specifies the owning exporting site from which input TC XML data is generated. This argument is mutually exclusive with the **-scope_rules** and **-scope_rules_mode** arguments.

The argument is not valid for bulk loading a briefcase file.

-transfermode

Specifies the transfer mode name that is to be used for import. If this argument is not specified, the utility uses the **TIEImportDefault** transfer mode. This argument is mutually exclusive with the **-scope_rules** and **-scope_rules_mode** arguments.

-optionset

Specifies the option set name that contains options to use during import. This argument is mutually exclusive with the **-scope_rules** and **-scope_rules_mode** arguments.

To import a briefcase file created using bulk extract, specify the option set with the **opt_bulk_load_bcz** value set to **TRUE**.

-scope_rules

Specifies that the input XML data contains scope rules, that is, transfer modes, closure rules, filter rules, property sets, actions rules, or transfer options sets. This argument is mutually exclusive with the **-site**, **-transfermode**, and **-optionset** arguments. See *Restrictions*.

-scope_rules_mode

Specifies the import behavior when an imported rule already exists in the database. If set to **ignore**, the rule is not imported. If set to **overwrite**, the existing database rule is overwritten. If you do not specify this argument, **ignore** behavior is used. If you do not specify the **-scope_rules** argument, this argument is invalid.

-requiredLang

Specifies a list of comma separated locale values. This list is used to ensure that localized attributes in the imported data have at least one representation that can be used as the attribute primary language at the importing site. It also defines a priority order for the exporter to determine the attribute primary language. The valid locale values must match the Java locale naming convention that consists of two groups of two-character identifiers separated by an underscore character (`_`) for a particular combination of language and region. For example, **zh_CN** represents Simplified Chinese in China and **en_US** represents English in the United States.

-allowedLang

Specifies a list of comma separated locale values. This list is used to get additional representations for localized attributes in the imported data for use at the importing site. The valid locale values must match the Java locale naming convention that consists of two groups of two-character identifiers separated by an underscore character (`_`) for a particular combination of language and region. For example, **zh_CN** represents Simplified Chinese in China and **en_US** represents English in the United States.

-low_level

Performs a fast import using POM level APIs. Fast import is normally used for consolidation of site databases. You must specify only the **-file** argument; all other arguments are ignored during fast imports.

This argument provides access to site consolidation functionality and requires setting a license key value for the **SITCONS_AUTH_KEY** environment variable. Open a support case on Support Center to get this license key.

If you import 4th Generation Design (4GD) data, you must run the **appmodel_fix_scope** utility at the target site to ensure 4GD data is represented properly.

-bulk_load

Performs a fast import of legacy data from a file containing low-level TC XML formatted data or a low-level TC XML briefcase file. You must specify only the **-file** and **-site** arguments; all other arguments are ignored when bulk loading low-level data. The **-site** argument is not valid for bulk loading a briefcase file.

Be aware of the following items:

- If the **-file** argument specifies a briefcase (.bcz) file, the utility switches to low-level briefcase import mode. If you specify any optional arguments (with the exception of the **-optionset** argument) when importing a briefcase file, they are ignored. If you do not specify the **-optionset** argument, the utility uses the **BulkLoadDefault** transfer option set. If you provide a transfer option set, it must contain the **opt_bulk_load_bcz** option set to **TRUE**.
- The import uses the volume specified in the TC XML if the **TIE_Volume_Map** preference is not set or does not specify a volume mapping for the volume named in the TC XML.

If the **TIE_Volume_Map** preference specifies a volume mapping for the volume named in the TC XML, the import uses the mapped volume. If the mapped volume is unavailable, the volume specified by **opt_use_default_volume** is used. If **opt_use_default_volume** is not set to **true**, an error is returned.

-tcfile_import

Imports **ImanFile** objects using a low-level TC XML formatted GSID-based file. This argument is valid only with the **-low_level** or **-bulk_load** arguments.

-islands

Imports one or more specific islands identified by *island-list*. **-islands** is valid only with the **-low_level** and **-bulk_load** arguments. Previous islands in the TC XML file (with higher ID values) must already exist in the database for an island to be imported successfully.

island-list is a comma-delimited list of island IDs or ranges of IDs, such as "10,8,7-5,3-". If no closing of a range is provided, all of the remaining islands in the TC XML file are imported.

-bypassSiteCheck

If specified, this switch bypasses the check that prevents the utility from updating objects at the same site and the check that prevents the update of replica objects.

Note:

If you are using this utility in a Multi-Site environment, run the utility at each of the sites to update the objects belonging to the respective sites.

Users have the capability of performing updates on replica objects, eliminating the need to re-replicate all updated objects.

Use this argument only with the **-bulk_load** argument for bulk update of attributes of local objects and import of archived audit records. Because this argument requires the **-bulk_load** argument that provides access to site consolidation functionality, **SITCONS_AUTH_KEY** must be set to a license key value.

-session_options

Specifies a comma-separated list of option name-value pairs. Names and values are delimited by colons, for example:

```
ContinueOnError:TRUE,GenerateReport:TRUE
```

-session_options_file

Specifies an ASCII file containing session option settings using option name-value pairs. Use this option instead of the **-session_options** option when you are reusing long lists of session option name-value pairs and when values contain system-reserved special characters.

Place one name-value pair on each line of the file as follows:

```
option_1:value_1
option_2:value_2
.
.
.
option_n:value_n
```

Set an option only once in the file.

-briefcase

Specifies the imported file is formatted as a briefcase file. The file is specified with the **-file** option.

Briefcase files exported using **-bulk_extract** are not supported by **-briefcase**.

-dryrun

When importing a low level briefcase file, specifies that a simulated import be run and the imported TC XML be validated. Warnings and errors in the TC XML are reported in the file specified by **-file** (with a *.html* extension).

If **-dryrun=validateXMLBeforeXslt** is specified, the exported TC XML is validated before the XSLT is applied to it. The briefcase file is not imported in either case. **-dryrun** cannot be used with the **-validate** option.

-validate

When importing a low level briefcase file, specifies that the imported TC XML be validated. Two levels of validation occur:

- In the first validation level, configuration aspects of the briefcase file such as the user, transformation rules, data types, and the target site are validated.
- In the second pass, the objects in the briefcase file are validated.

If errors are found in the first pass, the briefcase file is not imported. If errors are found only in the second pass, the briefcase file is imported.

Whenever warnings or errors are found, they are reported in the file specified by **-file** (with a *.html* extension).

If **-validate=validateXMLBeforeXslt** is specified, the internal (pre-export) TC XML is validated instead of the exported TC XML. **-validate** cannot be used with the **-dryrun** option.

-h

Displays help for this utility.

ENVIRONMENT

As specified in the *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

- When importing transfer option sets, if a local option set at the exporting site is specified for import, the utility assigns the local site as the site reference and the imported option becomes local to the importing site. If a remote option at the exporting site is specified for import, the site reference of that option set is expected to exist in the database at the importing site. If it is not, the import fails. To avoid this failure, you must manually create the option set before attempting the import.
- Not all Teamcenter data is supported for import. For a list of objects that are supported, see *Teamcenter Data Exchange*.

EXAMPLES

- The following example imports objects specified by the **-file** argument:

```
tcxml_import -file=file_name.xml
```

- The following example imports objects specified by the **-file** argument according to the rules given in transfer mode specified by the **-transfermode** argument:

```
tcxml_import -file=xml-file-name -transfermode=transfer-mode-name
```

- The following example imports objects specified by the **-file** argument and uses the value specified by the **-site** argument as the exporting site for this import:

```
tcxml_import -file=file_name.xml -site=site_ID
```

- The following example imports objects specified by the **-file** argument and uses the value specified by the **-xsl** argument to apply transformation on the input XML file:

```
tcxml_import -file=file_name.xml -xsl=xsl_file
```

- The following example imports objects specified by the **-file** argument and uses the value specified by the **-errorcontinue** argument to determine whether to roll back the import if an error occurs:

```
tcxml_import -file=file_name.xml -errorcontinue=yes
```

- The following example imports objects specified by the **-file** argument and uses the value specified by the **-optionset** argument to access the options that must be used during import:

```
tcxml_import -file=file_name.xml -optionset=option_set_name
```

- The following example imports objects specified with the **-file** argument with localizable attribute values in **en_US** and **fr_FR** locales:

```
tcxml_import -file=file_name.xml
-requiredlanguage=en_US -allowedlanguage=fr_FR
```

- The following example imports archived audit records from the file *AuditExport.xml*:

```
tcxml_import -bulk_load -bypassSiteCheck -file=c:/temp/AuditExport.xml
```

- The following example sets session option values specified in the external *sess_opts.cfg* file and imports **ImanFile** objects with physical files from the volume in the specified path:

```
tcxml_import -bulk_load -tcfile_import
-session_options_file=c:\config\sess_opts.cfg
-file=c:/temp0001.xml -source_volume=c:\temp\source_volume
```

- The following example explicitly sets session option values and imports objects with the **-file** argument with the specific island IDs:

```
tcxml_import -file=exportitem.xml -islands=10,8,7-5,3-
-session_options=opt_traverse_ref_org:false,
objsForOwnXfer:1857876201:wjUx4xecAAgcRA
```

- The following example simulates a low-level briefcase import, validating the imported low-level TC XML and generating an error report (and no briefcase file).

```
tcxml_import -file=d:\Temp\report.html
-optionset=TIEConfiguredLLBCZExportDefault -briefcase -dryrun
```

- The following example simulates a low-level briefcase import, validating the internal low-level TC XML and generating an error report without importing the briefcase file.

```
tcxml_import -file=d:\Temp\report.html
-optionset=TIEConfiguredLLBCZExportDefault
-briefcase -dryrun=validateXMLBeforeXslt
```

- The following example performs a low-level briefcase import, validating the exported TC XML and importing the briefcase file if no warnings or errors are encountered. If errors or warnings are encountered, a report file is created.

```
tcxml_import -file=d:\Temp\briefcase.bcz  
-optionset=TIEConfiguredLLBCZExportDefault -briefcase -validate
```

tcxml_xfer_ownership

DESCRIPTION

Transfers the ownership of a set of Teamcenter objects. You can use this utility to change ownership of all objects owned by a site during site consolidation, change ownership of related objects during transition from Teamcenter Enterprise to Teamcenter (flip-the-switch ownership transfer), or change object ownership when importing objects into Teamcenter from another data management system.

For performance reasons during site consolidation, this utility changes ownership of objects using SQL calls, not the POM API. Rich client refresh works only when POM calls are used to change an attribute, including ownership changes. Therefore, ownership changes that the utility enacts during an active rich client session are not reflected in the interface. You must restart all active rich clients to get the changes. This is normally not an issue because user access must be prohibited during the *time critical period* when this utility is used.

A valid license key is required to use this utility for site consolidation ownership changes.

The **perform_highlevel** argument enables the flip-the-switch ownership transfer feature of this utility. By default, during flip-the-switch ownership transfer, this utility uses the POM API to change object ownership. Therefore, ownership changes that the utility enacts during an active rich client session are reflected in the interface. You are not required to restart the active rich clients to get the changes.

For better performance, you can set the **TIE_flip_using_SQL** preference to **true** to use SQL calls during flip-the-switch ownership transfers. If you use this preference setting, you must restart all active rich clients to get the changes.

A valid license key is not required to use this utility when you enable the flip-the-switch feature.

SYNTAX

```
tcxml_xfer_ownership -u=user-name {-p=password | -pf=password-file} [-g=group]
  -action={extract | perform | update_status |
perform_highlevel | perform_highlevel_dryrun}
  [-inputfile=file-with-object-list]
  [-change_ownership_to=site-ID]
  [-dryrun [-startDate=MM/DD/YYYY -endDate=MM/DD/YYYY ] [-source_extinct] ]
  [-file=status-output-file]
  [-report=report-file-name]
  [-h]
```

ARGUMENTS

-u

Specifies the user ID.

This is a user with Teamcenter administration privileges.

When Security Services single sign-on (SSO) is enabled for your server, the **-u** and **-p** arguments are authenticated externally through SSO, rather than being authenticated against the database. If you do not supply these arguments, the utility attempts to join an existing SSO session. If no session is found, you are prompted to enter a user ID and password.

-p

Specifies the user's password.

This argument is mutually exclusive with the **-pf** argument.

-pf

Specifies the password file. The file must be a single-line ASCII file containing the password in clear text. Teamcenter Environment Manager prompts you for a password and creates the password file during installation.

This argument is mutually exclusive with the **-p** argument.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is assumed.

-action

Specifies whether to extract, perform, or update status.

To use the **extract**, **perform**, or **update_status** values, the **SITCONS_AUTH_KEY** environment variable must be set to a valid key value. Open a support case on Support Center to get this license key.

extract

Generates the file for ownership transfer.

The **-change_ownership_to**, **-report**, or **-dryrun** arguments are supported for this action.

perform

Transfers the ownership at target site.

The **-inputfile**, **-file**, **-source_extinct**, or **-dryrun** arguments are supported for this action.

update_status

Updates the status at source site.

The **-inputfile**, **-change_ownership_to**, **-source_extinct**, **-report**, or **-dryrun** arguments are supported for this action.

perform_highlevel

Performs a high-level (flip-the-switch) ownership transfer at the target site

The **-inputfile** and **-change_ownership_to** arguments are supported for this action. Any other arguments are ignored.

This action generates two outputs files, the *input-file-name_flip_objects.log* file lists all objects transferred and the *input-file-name_results.xml* file contains the flip status of objects given in input file.

perform_highlevel_dryrun

Generates a report to preview the ownership change of objects transferred by a high-level (flip-the-switch) ownership transfer at the target site.

The **-inputfile** and **-change_ownership_to** arguments are supported for this action. Any other arguments are ignored.

-inputfile

Specifies the name of the file containing the objects for ownership change.

-change_ownership_to

Specifies the site ID of the target site. This argument is required when the utility is run at the source site. If this argument is omitted, the utility uses the local site as the target site. If either the **-dryrun** or the **perform_highlevel_dryrun** argument is specified, the ownership transfer is not performed.

-dryrun

Generates a report to preview the ownership change of objects transferred by the site consolidation tools, and identifies inconsistencies.

To preview the objects at the target site that are owned by the source site and are inconsistent with the source site due to modifications made to them at the target site post-replication, set the **TIE_DRYRUN_VALIDATION=TRUE** preference at the target site.

The **-startDate** and **-endDate** arguments are valid only with the **-dryrun** argument. The utility generates a dry run report for objects transferred within the date range specified by the **-startDate** and **-endDate** values.

To use the dry run mode to check objects of an island owned by the source site that are not covered in the accountability table for ownership transfer, set the **TIE_DRYRUN_VALIDATION** preference to **TRUE** at the target site.

For debug information, set the following environment variables:

```
TC_SLOW_SQL=1
```

```
TC_SQL_DEBUG=BJPT
```

-startDate

Specifies the starting date for replicas and inconsistencies reported by the **-dryrun** argument. This argument is used in conjunction with the **-endDate** argument and is ignored if the **-dryrun** argument is not specified. The date supplied must be in supplied in a *MM/DD/YYYY* format.

-endDate

Specifies the ending date for replicas and inconsistencies reported by the **-dryrun** argument. This argument is used in conjunction with the **-startDate** argument and is ignored if the **-dryrun** argument is not specified. The date supplied must be in supplied in a *MM/DD/YYYY* format.

-source_extinct

Generates the list of replica items at the target site which are owned by the source site and which were not transferred by site consolidation tools.

This argument can be used only when the **-dryrun** argument is used and is ignored otherwise.

-file

Specifies the name of the output file (with an absolute path) that contains the status of ownership transfer of the objects at the target site. This argument is valid only for the site consolidation **-action=perform** argument.

-report

Specifies the name of the report file or extract file. This argument is valid only for site consolidation actions.

ENVIRONMENT

As specified in *Teamcenter Utilities*.

FILES

As specified in the *Teamcenter Utilities*.

RESTRICTIONS

None.

EXAMPLES

- To run at the source site and extract to a file the objects for ownership transfer:

```
tcxml_xfer_ownership -u=Tc-admin-user -p=password -g=group
-action=extract -report=D:\Temp\ownership_transfer_extract.txt
-change_ownership_to=target_site_id [-dryrun]
```

- To transfer the ownership or validate the inconsistencies for ownership transfer at the target site:

```
tcxml_xfer_ownership -u=Tc-admin-user -p=password -g=group
-action=perform -inputfile=ownership_transfer_extract.txt
-file=ownership_transfer_status.txt [-dryrun] [-source_extinct]
```

- To update the ownership transfer status or to generate report for a dry run at the source site:

```
tcxml_xfer_ownership -u=Tc-admin-user -p=password -g=group
-action=update_status -inputfile=ownership_transfer_status.txt
-report=ownership_transfer_update_status.txt
-change_ownership_to=target_site_id [-dryrun]
```

- To use flip-the-switch ownership transfer to change ownership at the target site:

```
tcxml_xfer_ownership -u=Tc-admin-user -p=password -g=group-
action=perform_highlevel -inputfile=flip_the_swich_input.txt
-change_ownership_to=target_site_id
```

- To use flip-the-switch ownership transfer to determine ownership change at the target site without actually changing ownership:

```
tcxml_xfer_ownership -u=Tc-admin-user -p=password -g=group
-action=perform_highlevel_dryrun -inputfile=flip_the_swich_input.txt
-change_ownership_to=target_site_id
```

- The following example shows the contents of a sample file used as the input for an ownership transfer (-inputfile=flip_the_switch_input.txt):

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- flip_the_switch_input.txt file -->
<TcFlipXML
  author="super user" date="2010-03-24" time="16:10:50" language="en"
  schemaVersion="2.0" >
<GSIdentity class="Cmponent" context="" factor="" atomic="false"
  label="udzjduesvsun0sugms2--aRZ" split_token="" sub-label="" system="213456789"
  transient_island_id="udzjdvcsvsun0sugms2--aRZ" elemId="1" />
<GSIdentity class="CmpnMstr" context="" factor="" atomic="false"
  label="udzjdvcsvsun0sugms2--aRZ" split_token="" sub-label="" system="213456789"
  transient_island_id="udzjdvcsvsun0sugms2--aRZ" elemId="2" />
<GSIdentity class="Assembly" context="" factor="" atomic="false"
  label="udzjdwksvsun0sugms2--aRZ" split_token="" sub-label="" system="213456789"
  transient_island_id="udzjdwksvsun0sugms2--aRZ" elemId="3" />
<GSIdentity class="AssmMstr" context="" factor="" atomic="false"
  label="udzjdxisvsun0sugms2--aRZ" split_token="" sub-label="" system="213456789"
  transient_island_id="udzjdwksvsun0sugms2--aRZ" elemId="4" />
</TcFlipXML>
```

You can construct the input file manually by copying the **GSIdentity** elements from the TC XML files used for importing the objects.

B. Consolidation from Oracle to Microsoft SQL Server

Environment and general information

This example provides information about consolidating two sites that have differing data models due to differences in the installed Teamcenter features and/or custom data model extensions.

The example assumes an environment that allows some of the site consolidation processes to be simplified or ignored in favor of emphasizing and presenting other important information. The sample environment has the following simple characteristics:

- The sites involved are small test sites that can be easily consolidated in one execution increment.
- The sites are copies of nonproduction test sites so no data changes during the replication activities. That is, the original site is essentially a back up. No product data is changing so fast synchronizations are not required.
- Because the consolidation activities are strictly a training exercise and do not exist in an operating Multi-Site environment, the final cleanup of the production environment is not required or performed.

You must adjust for more the complex characteristic in your production environment.

Additionally, this example assumes that the sites and infrastructure meet the requirements of site consolidation and that you have performed all the preparations steps identified in the preparation phase topics and any other preparations defined by Siemens Digital Industries Software.

Additionally, the example assumes the following:

- `TC_ROOT=D:\sql_tc\tc_root`
- `TC_DATA=D:\sql_tc\tc_data`
- `Portal=D:\sql_tc\tc_root\portal\portal.bat`
- All data created from the move is created as `D:\sql_tc\sitcons_work`
- Strawberry Perl is installed (you cannot use the `perl.exe` installed with Teamcenter)

Preparation phase process example

Configure the source site

1. Create the accountability tables in Oracle. The following SQL script provides the tables and triggers:

```

/* Copyright 2009.
   Siemens Product Lifecycle Management Software Inc.
   All Rights Reserved. */

/* These tables and triggers should be installed by the Oracle user
   so that there won't be access issues while reading and writing the
   data to these tables */

/* Creates the tables and triggers for Site Consolidation */
create table ACCT_TABLE(
exp_obj_uid varchar2(15) PRIMARY KEY,
led date,
island_anchor_uid varchar2(15),
state NUMBER);

create table SCRATCH_TABLE(
puid varchar2(15),
lsd date,
trigger_condition NUMBER);

CREATE INDEX "SCRATCH_TABLE_INDEX" ON "SCRATCH_TABLE" (puid);

create or replace trigger fast_sync_add_trigger
before insert on PPOM_OBJECT
referencing new as newRow
for each row
BEGIN
insert into scratch_table values (:newRow.puid, :newRow.plsd, '8');
END;
/

create or replace trigger fast_sync_delete_trigger
after delete on PPOM_OBJECT
referencing old as oldRow
for each row
BEGIN
insert into scratch_table values (:oldRow.puid, :oldRow.plsd, '9');
END;
/

```

2. Install the transfer modes:

```

tcxml_import -u=tcdba -p=tcdbapw -g=dba -scope_rules -scope_rules_mode=overwrite
-file=%tc_data%\siteConsolidationTransferModes.xml
tcxml_import -u=tcdba -p=tcdbapw -g=dba -scope_rules -scope_rules_mode=overwrite
-file=%tc_data%\siteConsolidationLWTransferModes.xml
tcxml_import -u=tcdba -p=tcdbapw -g=dba -scope_rules -scope_rules_mode=overwrite
-file=%tc_data%\siteConsolidationInternalClosureRules.xml

```

3. Extract shadow volume data from source site:

```
set sitcons_auth_key=XXXXXXXXXXXXXXXXXXXX
sitcons_extract_shared_vols -u=tcdba -p=tcdbapw -g=dba
-outputfile=D:\sql_tc\sitcons_work\source_fms_volumes.txt -prefix=source_
-fsc_id=FSC_source_ -node=localhost
```

4. Export site definition data required at the target database from source site:

```
admin_data_export -u=tcdba -p=tcdbapw -g=dba
-adminDataTypes=Organization -inputCriteria=POM_imc{name=*}
-outputPackage=D:\sql_tc\sitcons_work\site.zip
admin_data_export -u=tcdba -p=tcdbapw -g=dba
-adminDataTypes=Organization -inputCriteria=User{user_id=*}
-outputPackage=D:\sql_tc\sitcons_work\user.zip
admin_data_export -u=tcdba -p=tcdbapw -g=dba
-adminDataTypes=Organization -inputCriteria=Person{user_name=*}
-outputPackage=D:\sql_tc\sitcons_work\person.zip
admin_data_export -u=tcdba -p=tcdbapw -g=dba
-adminDataTypes=Organization -inputCriteria=Group{name=*}
-outputPackage=D:\sql_tc\sitcons_work\group.zip
admin_data_export -u=tcdba -p=tcdbapw -g=dba
-adminDataTypes=Organization -inputCriteria=Role{role_name=*}
-outputPackage=D:\sql_tc\sitcons_work\role.zip
```

5. Export the required workflow templates.

6. Package the export sites data model extensions into a Business Modeler IDE template project.

Do not include parts of the data model, for example access controls list (ACL), revision rules, display rules, preferences, and so forth, if they do not need to be reconciled. Ensure you have completed the tasks required to verify all necessary data is in place.

Configure the target site

1. Import source volumes:

```
set sitcons_auth_key=valid-supplied-key-value
sitcons_gen_shared_vols -u=tcdba -p=tcdbapw -g=dba
-file=D:\sql_tc\sitcons_work\source_fms_volumes.txt
-override_mode=merge
```

2. Install the transfer modes and the required site definition data:

```
admin_data_import -u=tcdba -p=tcdbapw -g=dba
-inputPackage=D:\sql_tc\sitcons_work\sites.zip -adminDataTypes=Organization
-mergeOption=keep_target
admin_data_import -u=tcdba -p=tcdbapw -g=dba
-inputPackage=D:\sql_tc\sitcons_work\user.zip -adminDataTypes=Organization
-mergeOption=Organization:keep_target
admin_data_import -u=tcdba -p=tcdbapw -g=dba
-inputPackage=D:\sql_tc\sitcons_work\person.zip -adminDataTypes=Organization
-mergeOption=Organization:keep_target
admin_data_import -u=tcdba -p=tcdbapw -g=dba
-inputPackage=D:\sql_tc\sitcons_work\group.zip -adminDataTypes=Organization
-mergeOption=Organization:keep_target
```

```
admin_data_import -u=tcdba -p=tcdbapw -g=dba
-inputPackage=D:\sql_tc\sitcons_work\role.zip -adminDataTypes=Organization
-mergeOption=Organization:keep_target
```

Using this order gives you more flexibility for setting default volumes. You can choose between the shared volumes or existing volumes and the target site.

3. Import the required workflow templates from the source site.
4. Using Teamcenter Environment Manager (TEM), add any additional Teamcenter features required to support the source site data model.

You can compare the source site and target **configuration.xml** files to determine any features that must be added. The **configuration.xml** file is located in the **install** directory under the **TC_ROOT** directory, for example:

```
<installed feature="42791A48DCBDD23FF08668F5847C96CC" name="JtUtilities Support" />
<installed feature="A22E1FC6EA1C4824BED64B2115B60FA5" name="MTM Data Card" />
<installed feature="E9B8A3F6B7B53AE540DCC54C13B3C201" name="Change Management/Data
Model" />
<installed feature="49A6EDA79E840AE24AC5CBC86718A4BC" name="Cmtmes/Data Model" />
<installed feature="F40CD40748BB470C98798DB9E294782A"
  name="Customization for Process Simulate Integration/Data Model" />
<installed feature="F3A6C89E5F9211D9A9C3810C1B1FBC64"
  name="Customization for eM-Server Integration/Data Model" />
<installed feature="AB97AAAC46471B7D5DC94C8A46E528ED" name="Issue Management/Data
Model" />
<installed feature="E4FEEA21B65873535FB16A137F7188C5" name="eBOP Reports
Customization/Data Model" />
<installed feature="DB57677C9C6456C44F9E1B3F260A0BAA" name="Workflow to Scheduling
Integration" />
<installed feature="49DB4E82CF098BE332C83AB78122E5B8" name="Database Daemons Action
Manager Service" />
<installed feature="50DB3E81CB098FE302D83EF78162E5D8"
  name="Database Daemons Subscription Manager Service" />
<installed feature="15CB3E81DB098FE311D83EE78162E58D" name="Database Daemons Task
Manager Service" />
<installed feature="51AB3E11DB083FE3D1283FE78662E58D"
  name="Database Daemons Tessellation Manager Service" />
<installed feature="3645D0D80B399B9544D9642F0292347C" name="NX Manager for Rich
Client" />
<installed feature="4FF970313813975F79306DF81D3C4A6B" name="Dispatcher Client for
Rich Client" />
<installed feature="301F12B06B3EAAB18DC0C2B9444F3B20"
  name="Translation Service Database Module/Data Model" />
<installed feature="C1F107F376D71271DE7B88D14CDC01AB" name="Dispatcher Server" />
<installed feature="301F12B06B3EAAB18DC0C2B9444F3B18" name="Dispatcher Client" />
<add feature="B2908BE2698228795A47BF26298E5F06" name="junk/Data Model" />
<spf feature="1A4C20CFAB9F7C7D6FFA9B0D9903877D" name="Customization for eM-Server
Integration" />
<spf feature="CFF57DB169F3DAAEB91DD16EAECA2A9E0" name="eBOP Reports Customization" />
<spf feature="9024ED922228151DDE54443BE044AF53" name="Issue Management" />
<spf feature="426293E52803AD96D8311536C8BBEC17" name="Cmtmes" />
<spf feature="326CA27DCABEFBB96D2735397986C1B9" name="Translation Service Database
Module" />
<spf feature="38E2829E9725CEBCC4C1156D61633B09" name="junk" />
```

```
<spf feature="DFCD67D128B68E6961C46E080B24B42D" name="Change Management" />  
<spf feature="77E9596F849365259472E29537BEE9C2"  
  name="Customization for Process Simulate Integration" />
```

5. Using TEM, install the template containing the source site extensions.

Prepare a transfer plan

1. Obtain product data item IDs from the Oracle database:
 - a. Log on to SQL.
 - b. Use the select command to get the **pitem_id** values from **pitem** objects.
 - c. Prepare the list by manually removing the header, footer, spaces, and column definitions.
2. Organize the list of items into files that can be used as input to **plm_report** utility and for replication. This may not be necessary if the site is sufficiently small. As a guideline, if the **plm_report** utility report indicates that transfers containing over 2 million objects in a single run may occur, it may be difficult to process and uncompress the data file and log files.

C. Shared FMS network

Prerequisites for a shared FMS network

The shared FMS network is a prerequisite for creating the shared volumes required by site consolidation. Before you begin to set up a shared FMS network the following must be true for both the source and target sites:

- You have installed the current version of Teamcenter.
- You have successfully configured Multi-Site.
- You can define and store datasets in Teamcenter from the rich client.

Key FMS configuration values before sharing

Following are the key FMS configuration values at the source site before configuring it for shared FMS.

Source site	Value
Name	SCSE1
Site ID	513202378
FSCID	FSC_cii3p070_ntpriv_scse1
FSCADDRESS	http://cii3p070:4544
FMSMASTERXML	FMSmaster_fsc_cii3p070_ntpriv_scse1.xml
FSCXML	FSC_cii3p070_ntpriv_scse1.xml
FMSmaster element	<code><fscmaster serves="true"></code>
fsc element	<code><fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544" ismaster="true"></code>

Following are the key FMS configuration values at the target site before configuring it for shared FMS.

Target site	Value
Name	SCSE2
Site ID	513205560
FSCID	FSC_cii3p103_ntpriv_scse2

Target site	Value
FSCADDRESS	http://cii3p103:4544
FMSMASTERXML	FMSmaster_fsc_cii3p103_ntpriv_scse2.xml
FSCXML	FSC_cii3p103_ntpriv_scse2.xml
FMSmaster element	<fscmaster serves="true">
fsc element	<fsc id="FSC_cii3p103_ntpriv_scse2" address="http://cii3p103:4544" ismaster="true">

Both sites have their FSCs configured as primaries. That is, each one is designated as the configuration source (repository) for FMS configuration information. This is defined by XML attribute values within the *FMSMASTERXML* and the *FSCXML* files.

The **SCSE1** site *FMSMASTERXML* file contains the following line:

```
<fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544"
ismaster="true">
```

The **SCSE1** site *FSCXML* file contains the following:

```
<fscmaster serves="true" />
<fsc id="FSC_cii3p070_ntpriv_scse1" />
```

The **SCSE2** site *FMSMASTERXML* file contains the following line:

```
<fsc id="FSC_cii3p103_ntpriv_scse2" address="http://cii3p103:4544"
ismaster="true">
```

The **SCSE1** site *FSCXML* file contains the following line:

```
<fscmaster serves="true" />
<fsc id="FSC_cii3p103_ntpriv_scse2" />
```

Therefore, each FSC is independent of the other. The source site (**SCSE1**) has no information about the target site (**SCSE2**) and the target site has no information about the source site except for the **multisiteimport** element that provides information that Multi-Site and Teamcenter require for transfers between the sites.

FMS configuration files before sharing

The entire contents of the *FMSMASTERXML* file for **SCSE1** (source) site before you configure it as a shared network:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE FMSworld SYSTEM "FMSmasterconfig.dtd">

  <!-- SCSE2 -->
  <multisiteimport siteid="513205560">
    <defaultfscimport fscid="FSC_cii3p103_ntpriv_scse2"
      fscaddress="http://cii3p103:4544" transport="lan"
      priority="0" />
  </multisiteimport>

<FMSworld>
  <FMSenterprise id="513202378">
    <fccdefaults>
      <property name="FCC_MaxExtentFileSizeMegabytes" value="256" overridable="true" />
      <property name="FCC_MaxExtentFiles" value="11" overridable="true" />
      <property name="FCC_HashBlockPages" value="6144" overridable="true" />
      <property name="FCC_MaximumNumberOfSegments" value="10688" overridable="true" />
      <property name="FCC_MaximumNumberOfFilePages" value="28672" overridable="true" />
      <property name="FCC_LogFile" value="$HOME/fcc.log|/tmp/$USER/fcc.log"
        overridable="true" />
      <property name="FCC_MaxReadCacheSize" value="1000M" overridable="true" />
      <property name="FCC_MaxWriteCacheSize" value="1000M" overridable="true" />
      <property name="FCC_CacheLocation" value="$HOME/SCSE1/FCCCache|/tmp/$USER/
FCCCache"
        overridable="true" />
    </fccdefaults>
    <fscgroup id="scse1_group">
      <fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544" ismaster="true">
        <volume id="1c0a49d243bb1e96d8ca" root="D:\\SCSE1\\scse1_vols\\volumel" />
        <transientvolume id="c915a12bead66b43d6c9c9d2044ac943"
          root="D:\\SCSE1\\transientVolume_ntpriv" />
      </fsc>
      <clientmap subnet="127.0.0.1" mask="0.0.0.0">
        <assignedfsc fscid="FSC_cii3p070_ntpriv_scse1" transport="lan" priority="0" />
      </clientmap>
    </fscgroup>
  </FMSenterprise>
</FMSworld>

```

The entire contents of the FMSMASTERXML file for SCSE2 (target) site before you configure it as a shared network:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE FMSworld SYSTEM "FMSmasterconfig.dtd">
<FMSworld>

  <!-- SCSE1 -->
  <multisiteimport siteid="513202378">
    <defaultfscimport fscid="FSC_cii3p070_ntpriv_scse1"
      fscaddress="http://cii3p070:4544" transport="lan"
      priority="0" />
  </multisiteimport>

  <FMSenterprise id="513205560">
    <fccdefaults>
      <property name="FCC_MaxExtentFileSizeMegabytes" value="256" overridable="true" />
      <property name="FCC_MaxExtentFiles" value="11" overridable="true" />
      <property name="FCC_HashBlockPages" value="6144" overridable="true" />
      <property name="FCC_MaximumNumberOfSegments" value="10688" overridable="true" />
    </fccdefaults>
  </FMSenterprise>
</FMSworld>

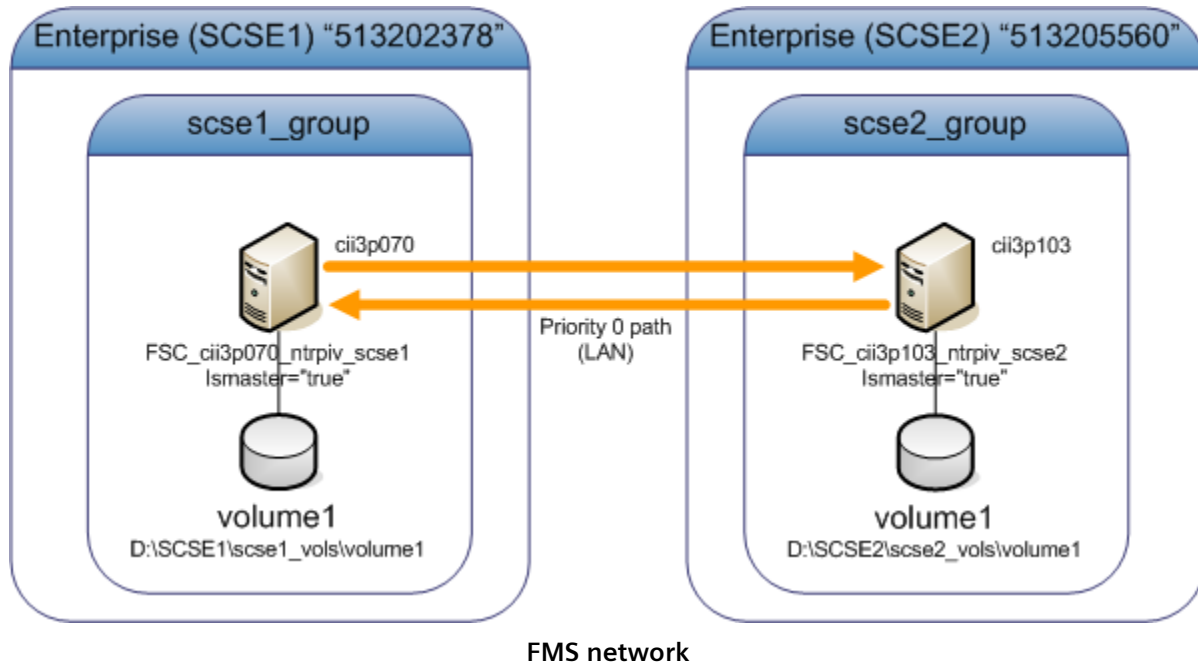
```

```

<property name="FCC_MaximumNumberOfFilePages" value="28672" overridable="true" />
<property name="FCC_LogFile" value="$HOME/fcc.log|/tmp/$USER/fcc.log"
  overridable="true" />
<property name="FCC_MaxReadCacheSize" value="1000M" overridable="true" />
<property name="FCC_MaxWriteCacheSize" value="1000M" overridable="true" />
<property name="FCC_CacheLocation" value="$HOME/SCSE2/FCCCache|/tmp/$USER/
FCCCache"
  overridable="true" />
</fccdefaults>
<fscgroup id="scse2_group">
  <fsc id="FSC_cii3p103_ntpriv_scse2" address="http://cii3p103:4544"
    ismaster="true">
    <volume id="1c0a49d250101e96e538" root="D:\\SCSE2\\scse2_vols\\volume1" />
    <transientvolume id="6b09e484075f338b6299c2eeb83164f9"
      root="D:\\SCSE2\\transientVolume_ntpriv" />
  </fsc>
  <clientmap subnet="127.0.0.1" mask="0.0.0.0">
    <assignedfsc fscid="FSC_cii3p103_ntpriv_scse2" transport="lan" priority="0" />
  </clientmap>
</fscgroup>
</FMSenterprise>
</FMSworld>

```

The FMS network before you configure it as a shared network:



Key FMS configuration values after sharing

The source site becomes a secondary FSC site in the shared FMS network. The following table shows the key FMS configuration values at the source site after you configure it as a shared network.

Source site	Value (secondary FMS)
Name	SCSE1
Site ID	513202378
FSCID	FSC_cii3p070_ntpriv_scse1
FSCADDRESS	http://cii3p070:4544
FMSMASTERXML	FMSmaster_fsc_cii3p070_ntpriv_scse1.xml
FSCXML	FSC_cii3p070_ntpriv_scse1.xml
fscmaster element	<pre><fscmaster serves="false" address="http:// cii3p103:4544"></pre>
fsc element	<pre><fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544" ismaster="false"></pre>

The target site becomes the primary FMS site in the shared FMS network. The key configuration values do not change at the target site.

Target site	Value (primary FMS)
Name	SCSE2
Site ID	513205560
FSCID	FSC_cii3p103_ntpriv_scse2
FSCADDRESS	http://cii3p103:4544
FMSMASTERXML	FMSmaster_fsc_cii3p103_ntpriv_scse2.xml
FSCXML	FSC_cii3p103_ntpriv_scse2.xml
fscmaster element	<pre><fscmaster serves="true"></pre>
fsc element	<pre><fsc id="FSC_cii3p103_ntpriv_scse2" address="http://cii3p103:4544" ismaster="true"></pre>

Configure shared FMS

In the shared FMS network, the SCSE2 site is designated as the target site and is established as the primary FMS. Therefore it is the repository of configuration information for this network. The SCSE1 site is designated the source site and a secondary FSC. The following process outlines how you create the primary and secondary configuration in your shared FMS network.

1. Create a working directory containing subdirectories for the target and source files. For example:

```
mkdir d:\workdir\FMS_SHARED
cd d:\workdir\FMS_ASIS
mkdir scse1-source
mkdir scse2-target
```

2. Copy the XML configuration (*FMSMASTERXML* and *FSCXML*) files from your source and target sites to the appropriate directory. The following is an example of the directory listing:

```
+---scse1-source
|   FMSmaster_FSC_cii3p070_ntpriv_scse1.xml
|   FSC_cii3p070_ntpriv_scse1.xml
|
\---scse2-target
     FMSmaster_FSC_cii3p103_ntpriv_scse2.xml
     FSC_cii3p103_ntpriv_scse2.xml
```

3. Edit the *FMSMASTERXML* file in the **scse1-source** directory to specify the secondary FSC. For example, change:

```
<fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544"
ismaster="true">
```

to:

```
<fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544"
ismaster="false">
```

Edit the *FSCXML* file in the **scse1-source** directory to specify the secondary FSC. For example, change:

```
<fscmaster serves="true" />
```

to:

```
<fscmaster serves="false" address="http://cii3p103:4544" />
```

4. Edit the *FMSMASTERXML* file in the **scse2-target** directory to change the FSC to a primary and create the shared network, for example:

Remove the following **multisiteimport** element:

```
<multisiteimport siteid="513202378">
  <defaultfscimport fscid="FSC_cii3p070_ntpriv_scse1"
    fscaddress="http://cii3p070:4544"
  transport="lan">
```

```

        priority="0" />
</multisiteimport>

```

Add the following **FMSenterprise** element in the existing (subordinate to) **FMSenterprise** element:

```
<FMSenterprise id="513202378" volumestate="normal" />
```

Change the **fscgroup** element to:

```
<fscgroup id="scse_group">
```

Add the following **fsc** element to the existing (subordinate to) the **fscgroup** element:

```

<fscgroup id="scse_group">
  <fsc id="FSC_cii3p070_ntpriv_scse1"
    address="http://cii3p070:4544" ismaster="false">
    <volume id="1c0a49d243bb1e96d8ca"
      enterpriseid="513202378"
      root="D:\\SCSE1\\scse1_vols\\volume1"
      priority="0" />
    <volume id="201b4adf2c2d1e96d8ca"
      enterpriseid="513202378"
      root="D:\\SCSE1\\scse1_vols\\volume2"
      priority="0" />
    <transientvolume id="c915a12bead66b43d6c9c9d2044ac943"
      enterpriseid="513202378"
      root="D:\\SCSE1\\transientVolume_tcdba"
      priority="0" />
  </fsc>

```

Add the following **assignedfsc** element to the existing (subordinate to) the **fscgroup** element:

```

<clientmap subnet="127.0.0.1" mask="0.0.0.0">
  <assignedfsc fscid="FSC_cii3p103_ntpriv_scse2" transport="lan"
    priority="0" />
  <assignedfsc fscid="FSC_cii3p070_ntpriv_scse1" transport="lan"
    priority="1" />
</clientmap>

```

Install the shared network configuration files

Use the following procedure to install the shared FMS network configuration files. Complete this procedure before you run the **sitcon_extract_shared_vols** utility at the source site and the **sitcons_gen_shared_vols** at the target site.

1. Stop the FSCs at the source and target sites.

2. Create backup files of the FMSMASTERXML and FSCXML configuration files at the source and target sites.
3. Copy the FMSMASTERXML and FSCXML files to the `TC_ROOT/fsc` directory of the source site.
4. Copy the FMSMASTERXML file to the `TC_ROOT/fsc` directory of the target site. The FSCXML file has not been changed.
5. Start the target site FSC and review the log files for any errors. It is important that the target site FSC be started first as it is the primary site in the shared network.

Resolve any errors found and restart the FSC, iterating until all errors are resolved.

6. Start the source site FSC and review the log files for any errors.

Resolve any errors found and restart the FSC, iterating until all errors are resolved.

7. Validate the FSCs are communicating with each other, for example, run the following commands from both the source and target sites:

```
fscadmin -s http://cii3p070:4544./status
fscadmin -s http://cii3p103:4544./status
```

Ensure the commands' output shows successful communications in both directions.

8. Validate you can access existing files within the network by, for example, creating a dataset from either site in the rich client.

The FMSMASTERXML files for both the source and target site has the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE FMSworld SYSTEM "FMSmasterconfig.dtd">

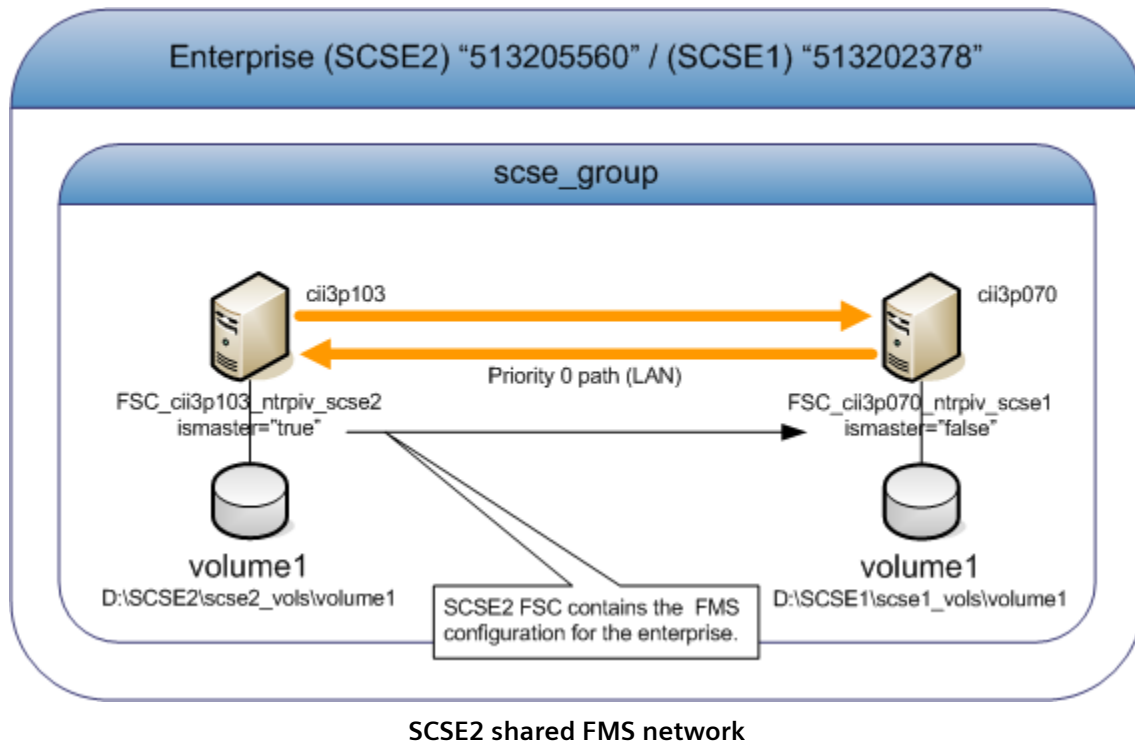
<FMSworld>
  <FMSenterprise id="513205560" volumestate="normal">
    <FMSenterprise id="513202378" volumestate="normal" />
    <fccdefaults>
      <property name="FCC_CacheLocation" value="$HOME/SCSE/FCCCache|/tmp/$USER/
FCCCache"
        overridable="true" />
      <property name="FCC_HashBlockPages" value="6144" overridable="true" />
      <property name="FCC_LogFile" value="$HOME/fcc.log|/tmp/$USER/fcc.log"
        overridable="true" />
      <property name="FCC_MaxExtentFileSizeMegabytes" value="256"
overridable="true" />
      <property name="FCC_MaxExtentFiles" value="11" overridable="true" />
      <property name="FCC_MaxReadCacheSize" value="1000M" overridable="true" />
      <property name="FCC_MaxWriteCacheSize" value="1000M" overridable="true" />
      <property name="FCC_MaximumNumberOfFilePages" value="28672"
        overridable="true" />
      <property name="FCC_MaximumNumberOfSegments" value="10688"
```

```

        overridable="true" />
</fccdefaults>
<fscgroup id="scse_group">
  <fsc id="FSC_cii3p070_ntpriv_scse1" address="http://cii3p070:4544"
    ismaster="false">
    <volume id="1c0a49d243bb1e96d8ca" enterpriseid="513202378"
      root="D:\\SCSE1\\scse1_vols\\volume1" priority="0" />
    <volume id="201b4adf2c2d1e96d8ca" enterpriseid="513202378"
      root="D:\\SCSE1\\scse1_vols\\volume2" priority="0" />
    <transientvolume id="c915a12bead66b43d6c9c9d2044ac943"
      enterpriseid="513202378"
      root="D:\\SCSE1\\transientVolume_tcdba" priority="0" />
  </fsc>
  <fsc id="FSC_cii3p103_ntpriv_scse2" address="http://cii3p103:4544"
    ismaster="true">
    <volume id="1c0a49d250101e96e538" enterpriseid="513205560"
      root="D:\\SCSE2\\scse2_vols\\volume1" priority="0" />
    <transientvolume id="6b09e484075f338b6299c2eeb83164f9"
      enterpriseid="513205560"
      root="D:\\SCSE2\\transientVolume_tcdba" priority="0" />
  </fsc>
  <clientmap subnet="146.122.10.1" mask="255.255.255.0">
    <assignedfsc fscid="FSC_cii3p070_ntpriv_scse1" priority="0" />
    <assignedfsc fscid="FSC_cii3p103_ntpriv_scse2" priority="1" />
  </clientmap>
  <clientmap subnet="146.122.62.1" mask="255.255.255.0">
    <assignedfsc fscid="FSC_cii3p103_ntpriv_scse2" priority="0" />
    <assignedfsc fscid="FSC_cii3p070_ntpriv_scse1" priority="1" />
  </clientmap>
  <clientmap subnet="146.122.200.1" mask="255.255.255.0">
    <assignedfsc fscid="FSC_cii3p103_ntpriv_scse2" priority="0" />
    <assignedfsc fscid="FSC_cii3p070_ntpriv_scse1" priority="1" />
  </clientmap>
</fscgroup>
</FMSenterprise>
</FMSworld>

```

After configuring the shared FMS network, it has the following configuration.



Create shared volumes

You must set up and verify the shared FMS network before you create the shared volumes. This is a requirement. Creating the shared volumes:

- Generates the shared volume definitions.
- Generates the **TIE_Volume_Map** preference map that indicates how the replication tools map volume references in the source site metadata when it is brought into the target site.
- Updates the shared FMS configuration file.

You use the **sitcons_extract_shared_vols** and **sitcons_gen_shared_vols** utilities to create the shared volumes.

1. From a Teamcenter enabled command shell at the source site, run the **sitcons_extract_shared_vols** utility, for example:

```
sitcons_extract_shared_vols -u=tcdba -p=tcdbapw -g=admingroup
-outputfile=scse1_sv_defs.txt -suffix=_scse1_scse2
-fsc_id=FSC_cii3p070_ntpriv_scse1
```

2. Move the output file to the target site.

3. From a Teamcenter enabled command shell at the target site, run the **sitcons_gen_shared_vols** utility, for example:

```
sitcons_gen_shared_vols -u=tcdba -p=tcdbapw -g=admingroup  
-file=scsel_sv_defs.txt -override_mode=merge
```

4. Validate the **TIE_Volume_Map** preference exists at the target site.

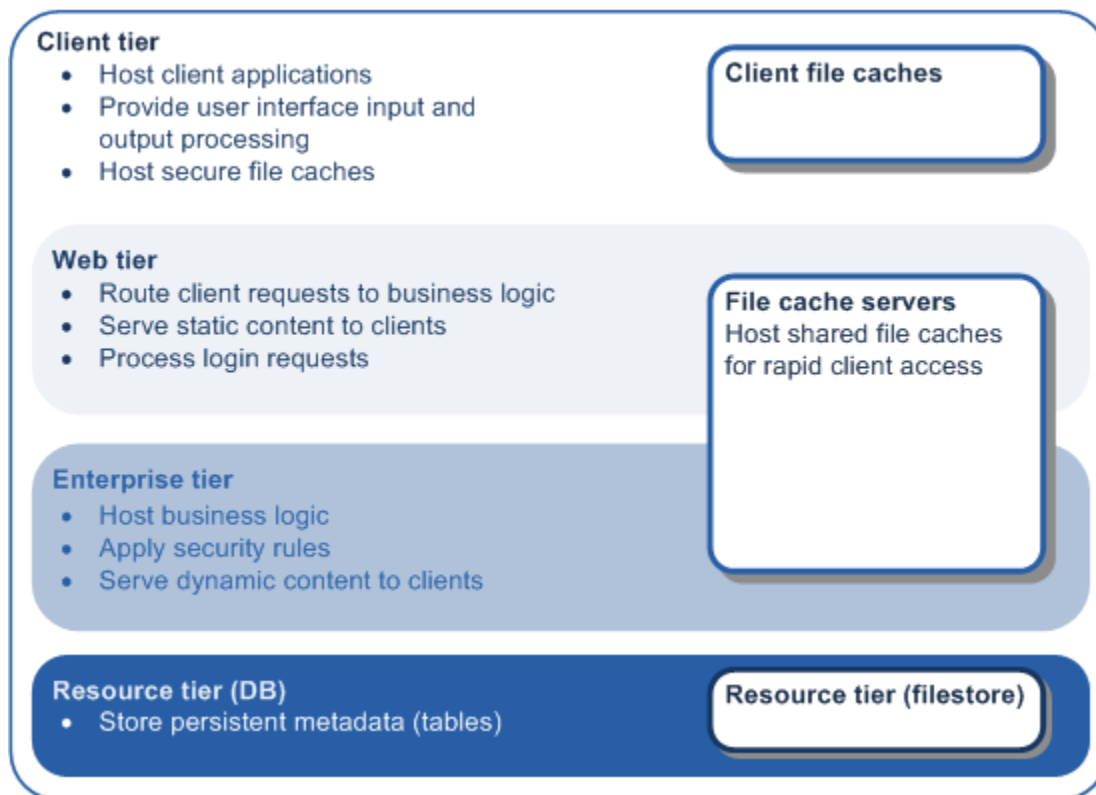
D. Auditing affected sites

Using audit components

Siemens Digital Industries Software recommends site audits to assess the readiness of the infrastructure for site consolidation.

Siemens Digital Industries Software can provide audit services to assist in the planning process for site consolidation, but the customer enterprise is responsible for the final plan and its execution.

Audit components help you to identify potential gaps in software and infrastructure that may impact the site consolidation process and the execution of the product after consolidation.



A complete audit looks at:

- Server configuration (review all tiers of the Teamcenter model).
 - Client
 - Web services
 - Enterprise tier

- Resources such as database servers, volume servers and FSC cache servers
- Network
 - Current topology definitions and any planned changes
 - Latency between sites
 - Current load
 - Planned growth
- Security
 - Requirements for clients being consolidated into the target site
 - Proxy/reverse proxy configurations
 - Firewall
 - Single sign-on (SSO) SSL
 - Supplier connectivity
- Software
 - Supported versions
 - Update requirements

Considering infrastructure issues

You are responsible for taking reasonable precautions to preserve the integrity of the data contained in the database and volumes servers.

- Backup and recovery

Data backup is generally part of the daily operations procedures established by the IT organization.

- Storage and compute capacity

Consider the need to support the increase in user storage, database storage, and compute capacity at the target sites.

Increases can include current capacity requirements along with expanded requirements for the additional users and the type of transactions to be incorporated.

- Subject matter experts

Teamcenter administrators and systems administrators can work to reduce any impact on the user base during the site consolidation process.

Changes in the environment, software upgrades, and outages are planned with the goal of site consolidation impacting the user community on a limited basis.

Recovering from failures during consolidation activities

Planning for recovery

The site consolidation process introduces changes to the underlying database, volume data, ODS and content in the Teamcenter software installation. During these activities, data can be impacted by events such as network outages or hardware failures. Such events may be out of the control of the IT organization at any time. A recovery plan must center on the aspects of environments affected by site consolidation activities during each of the three phases of site consolidation. The plan is not comprehensive of all aspects of recovery for the Multi-Site federation.

The site consolidation tools and process assume that user activity may continue to modify the existing environments and that the site-specific standard recovery process can address any user transactions that may be lost if a data snapshot must be restored.

To mitigate the impact of failures in the process, you should:

- Prepare a recovery plan for the site consolidation project.
- Retain log files and associated extract files for each phase. For a list of files, see *Log file names and locations*.
- When an execution increment is complete, archive and retain the log files and associated extract files until the process of site consolidation is complete for the current source and target sites.

Recovery activities

Recovery is the capability to roll back and restore the affected environments to a specific point in time.

Restoration involves standard backup practices that have been implemented by the IT organization.

Restoration includes:

- The database and all associated data files.

- The database software installation.
- The Teamcenter volume data.
- The Teamcenter software installation and all associated directories.

These activities are referred to as either a full snapshot or a data snapshot.

- A *data snapshot* refers to the database and Teamcenter data directory.

Key utility terms that can help you determine when to perform a data snapshot are: import, export, change, change ownership, transfer, replicate, confirm, move and update.

- A *full snapshot* is a copy of the entire set of data and software.

The target site, the source site and the ODS all need a full snapshot at the preparation phase, and then a data snapshot in the same time frame. This ensures any roll back is to the same time frame.

Creating a full snapshot and data snapshot can be a challenging logistical process but is necessary to provide for recovery in the event of a failure.

Preparation phase activities

The preparation phase is about data collection, in-depth analysis, and specific readiness steps to ensure site definition, product, and volume data are ready for consolidation.

During this phase, the majority of activity involves reading the content of the installation, analyzing the results, and then correcting any inconsistencies.

- Prior to any changes, a full snapshot of the environment should be made as a foundation recovery point. This would be the first roll back point.
- When the preparation phase is complete, make a data snapshot to reset the foundation of the data.
- If software is modified, a full snapshot is recommended so the current environment is restorable.

This becomes the next foundation recovery point.

Execution phase activities

The execution phase is performed in multiple increments that can be scoped based on the business drivers for the consolidation activity. Product data is moved during the this phase.

The typical structure of an execution phase increment consists of a replication period, followed by the time critical period (TCP) in which ownership change is performed for the product data that was replicated.

The execution phase has many steps that actively change the content of the environment.

- Perform this phase in multiple increments.
- During each increment, replicate one or more partitions of data and then reassign ownership from the source to the target site.
- Create a snapshot at each increment.

Replication, fast synchronization, and final preparation for the time critical period

Replication is the first step of any execution increment.

In replication, you use multiple steps for the fast source site export, mapping, and target site import of the product data as replicas. A data snapshot, along with standard backup procedures at the beginning of each time critical period, supports a larger recovery window in case roll back is needed.

The fast synchronization and final preparation steps are the last activities in this part of the process. If you need to redo any of the previous tasks, the saved log files and output from the execution activity can be reapplied. An analysis of what has changed and what can be reapplied is required for the correct reapplication of changes.

Time critical period

The time critical period is the period during which ownership of product data is changed from the source to the target site.

During this period the source and target site should be unavailable to production user.

- During the time critical period, all production users of the source and target systems must be logged off and the system must quiescent with respect to user activity. Use access is reestablished after the time critical period.

Perform a source and target data snapshot. This is the recovery point if anything needs to be recovered. Daily operations for backup should continue as part of the normal site activity.

- At the end of the time critical period, a data snapshot provides a new recovery point foundation for the consolidated sites that reflects the significant amount of data that is modified.

Cleanup and retirement

Cleanup and retirement refer to the final cleanup and decommissioning of the source site including activities such as volume data relocation, remaining ownership conflict resolution, and ODS configuration and cleanup.

The only recovery activities at this point are the ongoing daily operations for backup.

E. Performance tuning

Using database statistics

Statistics on the Oracle database must be current to support optimum performance from the indices. You should manually update the statistics using an **SQLplus** command form in Teamcenter:

```
SQL> exec dbms_stats.gather_schema_stats(ownname =>Tc-Oracle-  
user, estimate_percent  
=> 100, method_opt => 'FOR ALL COLUMNS SIZE  
1', degree=>8, cascade=>true);
```

If statistics have not been rebuilt recently, rebuilding may take an extended time to execute. Oracle provides an automated method of updating the statistics on a nightly basis through the Oracle enterprise manager.

The addition of indexes can produce dramatic improvements in the overall performance of the SQL as shown by the examples in *Index performance examples*. The examples show the results before and after adding indexes to the database. These were added after analyzing the performance of the database using the Oracle Enterprise Manager and the SQL tuner. Notice the improvement shown by the elapsed time to complete in each example.

Index performance examples

Run 1: wildcard (*) in command argument

Command syntax:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -item_id=AKS41559*  
-output_file=/apps1/user/scd1/data/AKS41559.bin  
-attributes_file=/apps1/user/scd1/17_April_2021/  
default_attributes_file.txt
```

Timing:

- Start time: 4/29/2021 16:00:09 EDT
- End time: 4/30/2021 03:20:47 EDT

Elapsed time: 11h 20m 38s

Run 2: no wildcard (*) in command argument

Command syntax:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -item_id=AKS41559  
-output_file=/apps1/user/scd1/data/AKS41559.bin  
-attributes_file=/apps1/user/scd1/17_April_2021/  
default_attributes_file.txt
```

Timing:

- Start time: 5/1/2021 14:15:27 EDT
- End time: 5/2/2021 01:04:19 EDT

Elapsed time: 10h 49m 6.64s

Run 3: no wildcard (*) in command argument, added db index for export records

Command syntax:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -item_id=AKS41559  
-output_file=/apps1/user/scd1/data/AKS41559.bin  
-attributes_file=/apps1/user/scd1/17_April_2021/  
default_attributes_file.txt
```

Timing:

- Start time: 5/4/2021 09:29:51 EDT
- End time: 5/4/2021 10:32:57 EDT

Elapsed time: 1h 3m 12.01s

Run 4: no wildcard (*) in command argument; db index for export records; added db index for variant expressions

Command syntax:

```
plm_report_extract -u=tc-admin-user -p=password -g=group -item_id=AKS41559  
-output_file=/apps1/user/scd1/data/AKS41559.bin  
-attributes_file=/apps1/user/scd1/17_April_2021/  
default_attributes_file.txt
```

Timing:

- Start time: 5/4/2021 13:11:39 EDT

- End time: 5/4/2021 13:13:20 EDT

Elapsed time: 0h 1m 45.86s

Setting Oracle SDU/TDU values

Set oracle SDU/TDU values to improve network performance for Oracle.

```
ORACLE_HOME/network/admin/tnsnames.ora
DESCRIPTION=
  (SDU=8192)    //service layer buffer size
  (TDU=8192)    //transport layer size
ORACLE_HOME/network/admin/listener.ora
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_DESC =
        (SDU=8192)
        (SID_NAME = my_sid)
        (ORACLE_HOME = c:\oracle\product\10.2.0)
      )
    )
  )
```

Setting the Oracle tcp.nodelay parameter

You can set the **tcp.nodelay** parameter in the **ORACLE_HOME/network/admin/sqlnet.ora** file.

By default, because Oracle*Net waits until the buffer is full before transmitting data, requests are not always sent immediately to their destinations. This is common when large amounts of data are streamed from one location to another, and Oracle*Net does not transmit the packet until the buffer is full.

Adding a **sqlnet.ora** file and specifying a **tcp.nodelay** to stop buffer flushing delays can sometimes improve this situation. The parameter can be used both on the client and server. The **sqlnet.ora** statement is:

```
tcp.nodelay = YES
```

When this parameter is specified, TCP buffering is skipped so that every request is sent immediately. Slowdowns in the network may be caused by an increase in network traffic due to smaller and more frequent packet transmission. This can be set on the client and the server.

Oracle System Global Area

The System Global Area (SGA) is a group of shared memory structures that contains data and control information for one Oracle database. The SGA is allocated in memory when an Oracle database instance is started.

For large databases, this value can range from 8 GB to 16 GB, and larger. Site consolidation executes additional SQL, and this puts additional demands on the buffer caches.

Review your current settings and determine if you have the resources necessary to increase this value if needed.

The following example reflects an 8 GB SGA, with **sga_max_size=8GB**, and with a breakdown of components. This setting is for a database with approximately 250 GB of tablespace. Automatic memory management is enabled by setting **db_cache_size =0** and **sga_target=sga_max**.

SGA component allocation (MB)

Shared pool	368
Buffer cache	7760
Large pool	16
Java pool	16
Other	32

Oracle Program Global Area

The Program Global Area (PGA) is a memory buffer that contains data and control information for a server process. A PGA is created by Oracle when a server process is started.

The sum of PGA and SGA should be less than the total system memory minus memory required by the operating system and other applications. The PGA setting in the example below is at 2 GB.

Determine if sufficient resources exist if you wish to increase the value. This setting is for a large database that is approximately 250 GB of tablespace. The parameter setting for this value is **pga_aggregate_target**.

PGA parameter values

Aggregate PGA target	2 GB
Current PGA allocated (KB)	88407
Maximum PGA allocated (KB)	140901 (since startup)
Cache hit percentage (%)	100

Additional Oracle parameters that impact database performance

filesystemio_options = setall

Specifies I/O operations for file system files.

This is used to support direct I/O for NFS.

optimizer_dynamic_sampling = 2

Controls the level of dynamic sampling performed by the optimizer.

This is to support Oracle 10.0.0 or later.

optimizer_index_caching = 95

Adjusts the behavior of cost-based optimization for nested loops joins and **IN-list** iterators.

You can modify the optimizer's assumptions about index caching for nested loops joins and **IN-list** iterators by setting this parameter to a value between 0 and 100 to indicate the percentage of the index blocks the optimizer should assume are in the cache. Setting this parameter to a higher value makes nested loops joins and **IN-list** iterators look less expensive to the optimizer. As a result, it is more likely to select nested loops joins over hash or sort-merge joins and to pick indexes using **IN-list** iterators over other indexes or full table scans. The default for this parameter is 0, which results in default optimizer behavior.

optimizer_index_cost_adj = 10

Tunes optimizer behavior for access path selection to be more or less likely to selecting an index access path over a full table scan.

The default for this parameter is 100 percent, which causes the optimizer to evaluate index access paths at the regular cost. Any other value makes the optimizer evaluate the access path at that percentage of the regular cost. For example, a setting of 50 makes the index access path look half as expensive as normal.

statistics_level = typical

Specifies the level of collection for database and operating system statistics.

The Oracle database collects these statistics for a variety of purposes, including making self-management decisions. The default setting of **TYPICAL** ensures collection of all major statistics required for database self-management functionality and provides best overall performance. The default value should be adequate for most environments.

timed_statistics = true

Specifies whether or not statistics related to time are collected

Automatic Database Diagnostic Monitor

The Automatic Database Diagnostic Monitor (ADDM) is provided with the Oracle Enterprise Manager. ADDM is a GUI implementation *statspack*.

This tool:

- Helps you analyze SQL statements.
- Suggests alternative execution plans

- Can be used to trend data across multiple time slices to get an up to review historical data and up-to-date performance statistics.

Networking

The network between sites that are being consolidated must be monitored for the current user load.

- Establish a baseline of existing utilization and performance.
- Use the baseline for comparing new traffic load to determine changes such as increased utilization.

Network latency can also contribute to performance slowdowns.

File system

The file system server and underlying protocol can have an impact on the I/O performance of the database.

- Site consolidation utilities can generate a significant amount of I/O generally in the form of reads. This I/O overhead is in addition to the existing I/O load that is currently generated by the users.
- If the file system that holds the Oracle tablespaces is an NFS mounted files system on NAS device, Oracle 10g utilizes its own internal caches for file I/O and looks to bypass the system cache set up by the OS. The Oracle **filesystemio_options = setall** parameter instructs Oracle to utilize the internal buffering.

Check with your hardware/operating system vendor regarding the correct setting for your Oracle server.

Site consolidation with Teamcenter running on Solaris

Solaris platforms were supported by earlier versions of Teamcenter. If you are performing site consolidation with sites using Solaris, tune I/O using the **forcedirectio** mount option on the **vfstab** file.

The **forcedirectio** performance option benefits only large sequential data transfers. The default behavior is **noforcedirectio**.

- If **forcedirectio** is specified, forced direct I/O is used for the duration of the mount.
- If the file system is mounted using **forcedirectio**, data is transferred directly between client and server, with no buffering on the client.
- If the file system is mounted using **noforcedirectio**, data is buffered on the client.

The following example is from a Solaris 10 system:

```
uscisnap004:/vol/vol1/na3/db3 - /db3 nfs - yes  
rw,vers=3,rsize=32768,wsiz=32768,proto=tcp,intr,hard,bg,forcedirectio
```


F. Using log files

Log file names and locations

The log files, such as the **syslog** files, are created in the directory specified by the **TC_TMP_DIR** environment variable, which is normally assigned to **/tmp** or **/var/tmp**. A **syslog** file has a name of **utility_namennnn.syslog** where **utility_name** is the name of the utility creating file, and **nnnn** is the process ID.

When log files are configured manually, the files can be directed to any specified location as shown in this partial script:

```
#!/bin/ksh
. /apps1/scripts/doenv
export TIE_DEBUG=2
export TC_KEEP_SYSTEM_LOG=true
mydir="/apps2/tcdba/drop3patch/corpparts_reportnorm/plmr_gl_lw_8_19/dir_11to15"
myfile="cp_gl_11to15"
echo "Run Directory "$mydir">>nohup.out
echo "File prefix "$myfile">>nohup.out
echo " "
echo "---STARTING PLM_REPORT EXTRACT,UNCOMPRESS RUN,tcxml_extract---">>nohup.out
echo " ">>nohup.out
echo "---Starting PLM_Report lightweight Closure Rule Extract-----">>nohup.out
mylogdir=$mydir/$myfile"_lwcr_runlogs"
mkdir $mylogdir
export TC_TMP_DIR=$mylogdir
mylwoutfile=$mydir/$myfile"_lwcr_ext"
echo "LWCR Extract file "$mylwoutfile">>nohup.out
echo "Log directory "$mylogdir">>nohup.out
echo "plm_report_extract -u=tcdba -p=tcdbapw -g=admingroup -verbose
-inputfile=$mydir/$myfile.txt -output_file=$mylwoutfile.bin
-attributes_file=$TC_DATA/default_attributes_file.txt
-optionset=SiteConsolidationLW">>nohup.out
date>>nohup.out
nohup plm_report_extract -u=tcdba -p=tcdbapw -g=admingroup -verbose
-inputfile=$mydir/$myfile.txt -output_file=$mylwoutfile.bin
-attributes_file=$TC_DATA/default_attributes_file.txt
-optionset=SiteConsolidationLW
wait
echo "Stopping LW Extract.....">>nohup.out
date>>nohup.out
...
```

Utility debug options and default file locations

Step	Utility	Debug options	Logs	Location
1	plm_report_extract	TIE_DEBUG	syslog Extract log	Defined location. Output file argument location.
2	plm_report_consistency_analysis	None	syslog	Defined location.

Step	Utility	Debug options	Logs	Location
3	<code>plm_report_constraint_analysis</code>	None	syslog	Defined location.
4	<code>tcxml_export</code>	TIE_DEBUG	syslog Intermittent logs at specified time intervals Export log	Defined location. Output file argument location. Log file are generated in the output file location.
5	<code>tcxml_import</code>	TIE_DEBUG	syslog Import and import_result log	Defined location. At the export XML location.
6	<code>tcxml_confirm_export</code>	None	syslog	Defined location.
7	<code>tcxml_xfer_ownership</code>	TIE_DRYRUN_VALIDATION	syslog	Defined location.
8	<code>sitcons_xfer_owner_mgr</code>	TIE_DRYRUN_VALIDATION	syslog Extract file generated at source site Status file generated after <i>Perform</i> step at target site Status file generated after update status at source site	Defined location. Source site transient volume. Target site transient volume. Source site transient volume.
9	<code>sitcons_replicate_mgr</code>	None	syslog Exporter logs and files Importer logs and files	Defined location. Source site transient volume. Target site transient volume.
10	<code>sitcons_fix_ixr</code>	None	syslog	Defined location.
11	<code>site_consolidation_accountability</code>	None	syslog	Defined location.
12	<code>sitcons_user_folders</code>	None	syslog	Defined location.
13	<code>sitcons_extract_shared_vols</code>	None	syslog	Defined location.
14	<code>sitcons_gen_shared_vols</code>	None	syslog	Defined location.

TIE_DEBUG environment variable

Based on the **TIE_DEBUG=0** environment variable value, the **tcxml** utility low-level export and the **plm_report_extract** utility traversal actions can generate log files and **syslog** output.

- `output_file_name.log`

A separate log file, generated in the output file location, containing the following:

- The number of traversed objects
- The PUID, island ID, predecessor UID, island anchor UID, and class name for each object

- `syslog`

The `syslog` file is located in a specified location.

Debug information is written to log files and `syslog` output based on the value of the `TIE_DEBUG` environment variable. When reporting issues to support engineers, provide the logs and `syslog` files generated with `TIE_DEBUG=2`.

- `TIE_DEBUG=0`

When not set, or if set to zero (0), the export log contains the PUID, island ID, class ID, predecessor UID, and island anchor UID information for traversed objects.

- `TIE_DEBUG>0`

The export log contains time-SQL-memory records for traversal and serialization batches.

```
***Traversal Record(Memory:SQL:Time:)**
Memory : 63141 ,SQL: 349 ,TIME: 5.770000s cpu, 25.435502s real
2 seconds to POM DB x-act
*** Serialize Record for Island Batch (Memory:SQL:Time:): 1 (Size:538)
Memory : 428909 ,SQL: 621 ,TIME: 3.520000s cpu, 16.179516s real
```

- `TIE_DEBUG=2`

This setting is for Siemens Digital Industries Software support engineers. It prints information to the `syslog` output, such as rule strings and cross-island clauses. The `syslog` output also contains detailed step-by-step execution of traversal information. For example, the `syslog` output contains entries for each input, clauses to be executed, and the results upon execution. The export log contains the count of traversed objects grouped by class.

- `TIE_DEBUG=4`

This setting is for users processing large items. The output includes intermittent logs at specified time intervals and traversed object count progress on the console and in the `syslog` file.

```
Set LOG_BATCH_SIZE=5000
//Writes to console & syslog for every 5000 increment in traversed object count.
Default '20000'
```

```

C:\WINDOWS\system32\cmd.exe
Traversed Object Count: 1252001 - Wed Aug 05 16:35:47 2009
Traversed Object Count: 1253001 - Wed Aug 05 16:35:50 2009
Traversed Object Count: 1254001 - Wed Aug 05 16:35:55 2009
Traversed Object Count: 1255001 - Wed Aug 05 16:35:56 2009
Traversed Object Count: 1256001 - Wed Aug 05 16:35:56 2009
Traversed Object Count: 1257001 - Wed Aug 05 16:35:56 2009
Traversed Object Count: 1258001 - Wed Aug 05 16:35:58 2009
Traversed Object Count: 1259001 - Wed Aug 05 16:35:58 2009
Traversed Object Count: 1260001 - Wed Aug 05 16:36:02 2009
Traversed Object Count: 1261001 - Wed Aug 05 16:36:12 2009
Traversed Object Count: 1262001 - Wed Aug 05 16:36:12 2009
Traversed Object Count: 1263001 - Wed Aug 05 16:36:12 2009

```

```

Set LOG_TIME_INTERVAL=300
//Writes out an intermittent log file every 300s (5 mins) with information on
objects traversed in this interval. Default '300'

```

Sample run of tcxml_import low-level

Run `tcxml_import` with the `-low_level` flag to generate key log files.

Input file:

`D:\Testing\items_set1.xml`

Command:

```
tcxml_import -u=tc-admin-user -p=password -g=group -file=D:\Testing\items_set1.xml -low_level
```

Output files:

- **items_set1_import_results.log**

Import results log.

- **items_set1_importer.log**

Import operation details log.

- **items_set1_pretraverse.log**

Pretraversal at target site log.

Import results log for <input-xml-filename>_import_results.log

The *input-xml-filename_import_results.log* contains the required input to be passed to the **tcxml_confirm_export** utility.

```
# TIE Fast Import Results of
D:\Testing\items_set1.xml
CrHpXutbh7iXbC 1 2048
CDMpmR3VxHqEoC 0 2048
```

There are three fields in each row:

- *UID*

UID of the principal object of the data island.

- *Import status*

1 means the import failed.

0 means the import succeeded.

- *Object mode*

Internal.

Import operation details log for <input-xml-filename>_import.log

Import operation details log	Comments
<pre>***** Fast Import Log File Header ***** Teamcenter version : Teamcenter P2007.1.8.2009072800 Syslog-FileName : C:\Temp\tcxml_import.exe135885e5. syslog Input File Name : D:\Testing\items_set1.xml *****</pre>	Header showing XML input file name and the output syslog file name.
<pre>Processing Island: 2 No Pre-traversed objects found for this Island Resolve Import Mode: Import Mode SrKpmR3VxHqEoC -> 1 Import Mode SnJpmR3VxHqEoC -> 1 Import Mode CDMpmR3VxHqEoC -> 1 Import Mode SrIpmR3VxHqEoC -> 1 Import Mode SrNpmR3VxHqEoC -> 1 Import Mode SrApmR3VxHqEoC -> 1 Import Mode CDPpmR3VxHqEoC -> 1 Import Mode S\$IpmR3VxHqEoC -> 1 Import Mode SrDpmR3VxHqEoC -> 1 Import Mode SPLpmR3VxHqEoC -> 1</pre>	<p>Processing of island starts.</p> <p>Import mode per UID object:</p> <p>1 – Object does not exist in target; to be created.</p> <p>2 – Object exists but out dated; to updated.</p> <p>3 – Objects exists with save date in sync; to be skipped.</p> <p>Processing sequence:</p>

Import operation details log

```

Loading Objects
Creating Objects
Creating Object SrKpmR3VxHqEoC for RevisionAnchor
Creating Object SnJpmR3VxHqEoC for ItemRevision
Creating Object CDMpmR3VxHqEoC for Item
Creating Object SrIpmR3VxHqEoC for Dataset
Creating Object SrNpmR3VxHqEoC for Dataset
Creating Object SrApmR3VxHqEoC for Form
Creating Object CDPpmR3VxHqEoC for Form
Creating Object S$IpmR3VxHqEoC for ImanRelation
Creating Object SrDpmR3VxHqEoC for ImanRelation
Creating Object SPLpmR3VxHqEoC for ImanRelation
Resolving Attribtues:
Setting Attribtues:
Saving Objects:
Num objects to Save: 10

Processing Island : 1
No Pre-traversed objects found for this Island
Resolve Import Mode:
Import Mode CrHpXutbh7iXbC -> 2
Import Mode CrNpXutbh7iXbC -> 2
Import Mode CvGpXutbh7iXbC -> 2
Import Mode CvNpXutbh7iXbC -> 2
Import Mode CrKpXutbh7iXbC -> 2
Import Mode CvDpXutbh7iXbC -> 2
Import Mode CvApXutbh7iXbC -> 2
Loading Objects
Creating Objects
Creating Object CrHpXutbh7iXbC for Item
Creating Object CrNpXutbh7iXbC for ImanRelation
Creating Object CvGpXutbh7iXbC for ImanRelation
Creating Object CvNpXutbh7iXbC for
ProjectObjectRelation
Creating Object CrKpXutbh7iXbC for Form
Creating Object CvDpXutbh7iXbC for Form
Creating Object CvApXutbh7iXbC for ItemRevision
Resolving Attribtues:
Setting Attribtues:
Setting Attributes for Item(CrHpXutbh7iXbC)
SetAttr owning_site = x7IpSEuDh7lSrA
SetAttr acl_bits = 0
SetAttr active_seq = 1
SetAttr archive_date =
Import Failed with Error: Invalid Date ( errcode
515032)
for Island: 2 (Principal Object:CrHpXutbh7iXbC)

Import Results Summary:
-----
Total number of Input Objects : 20
  Organization Ref Objects : 3
  Stub References : 0
  Objects for Import : 17
Total Number of Objects processed for Import: 17
(in 2 Islands)
Objects By Import Mode
  Number of Objects new : 10

```

Comments

– Pretraverse target objects.

– Find import mode.

– Load existing objects.

– Create replica objects.

The last two lines indicate island objects are successfully saved.

Processing of island starts.

The last line indicates the island failed to import.

Error message (error code) for the island ID with the principal object UID.

Import results summary starts.

Displays the total number of input import objects.

Displays the number of import objects that are to be newly created and that already exist.

Import operation details log**Comments**

```

Number of Object already exists : 7
Import Results
Number of Objects Imported successfully : 10
(in 1 Islands)
Number of Objects Import failed : 7
(in 1 Islands)
Number of Objects skipped for Import : 0

```

Displays the number objects in Islands succeeded, failed and skipped.

Orchestration data transfer log files

Data transfer is managed by the **sitcons_replicate_mgr** utility. A transaction ID is written to the Teamcenter server **syslog** file, visible on the Teamcenter Integration Framework activity status page.

All files are located on the target site transient volume. Files created by the Teamcenter Integration Framework functionality are written to folders named with unique strings, and file names that reflect the Teamcenter Integration Framework activity ID.

- Sample Teamcenter Integration Framework activity folder name:

```
000000000000202f4afac2971fa1c050
```

- Sample Teamcenter Integration Framework activity file name:

```
1257928391508-28
```

Step	Description	Folder/Files
1	Orchestration scheduling	–
2	Orchestration export	Folder: <i>transaction-ID_lltie_export_timestamp</i> Files: <i>transaction-ID_timestamp.log</i> <i>transaction-ID_timestamp.xml</i>
3	Orchestration data mapping	Folder: <i>unique_string</i> File:

Step	Description	Folder/Files
		<i>Teamcenter_integration_framework_activity_ID</i>
4	Orchestration FMS file movement to target site	File: <i>Teamcenter_integration_framework_activity_ID</i>
5	Orchestration import	Files: <i>transaction_ID_pretraverse.log</i> <i>transaction_ID_importer.log</i> <i>transaction_ID_import_results.txt</i>
6	Orchestration FMS file movement to source site	File: <i>transaction_ID_import_results.txt</i>
7	Orchestration confirm import	Folder: <i>transaction_ID_timestamp</i> File: <i>transaction_ID_ll_confirm_export.txt</i>

Orchestration change ownership log files

Ownership change is managed by the **sitcons_xfer_owner_mgr** utility. A transaction ID is written to the Teamcenter server **syslog** file, visible on the Teamcenter Integration Framework activity status page.

All files are located on the respective site transient volume in a folder named *transaction-ID_timestamp*. When the **-dryrun** argument is used, the output file name includes **_dryrun**.

Step	Description	Files
1	Get objects for ownership transfer at source site.	<i>transaction-ID_ll_xfer_owner_extract.txt</i> <i>transaction-ID_ll_xfer_owner_dryrun_extract.txt</i>
2	FMS file movement to target site.	<i>transaction-ID_ll_xfer_owner_extract.txt</i>

Step	Description	Files
		<i>transaction-ID_11_xfer_owner_dryrun_extract.txt</i>
3	Transfer ownership at target site.	<i>transaction-ID_11_xfer_owner_target.txt</i> <i>transaction-ID_11_xfer_owner_dryrun_target.txt</i>
4	FMS file movement to source site	<i>transaction-ID_11_xfer_owner_target.txt</i> <i>transaction-ID_11_xfer_owner_dryrun_target.txt</i>
7	Transfer ownership at source site.	<i>transaction-ID_11_xfer_owner_src.txt</i> <i>transaction-ID_11_xfer_owner_dryrun_src</i>

Teamcenter Integration Framework tables cleanup

Some step failures that occur when you use Teamcenter Integration Framework orchestration for replication or ownership transfer are stored in the Teamcenter Integration Framework tables for retry purposes. You can use the following scripts to remove the failed steps from the tables.

1.3.3.drop.sql script listing

```
-- Apache ODE - SimpleScheduler Database Schema
--
-- Apache Derby scripts by Maciej Szeffler.
--
--

drop table ODE_JOB cascade constraints purge ;
drop table BPEL_ACTIVITY_RECOVERY cascade constraints purge ;
drop table BPEL_CORRELATION_PROP cascade constraints purge ;
drop table BPEL_CORRELATION_SET cascade constraints purge ;
drop table BPEL_CORRELATOR cascade constraints purge ;
drop table BPEL_CORRELATOR_MESSAGE_CKEY cascade constraints purge ;
drop table BPEL_EVENT cascade constraints purge ;
drop table BPEL_FAULT cascade constraints purge ;
drop table BPEL_INSTANCE cascade constraints purge ;
drop table BPEL_MESSAGE cascade constraints purge ;
drop table BPEL_MESSAGE_EXCHANGE cascade constraints purge ;
drop table BPEL_MEX_PROPS cascade constraints purge ;
drop table BPEL_PLINK_VAL cascade constraints purge ;
drop table BPEL_PROCESS cascade constraints purge ;
drop table BPEL_SCOPE cascade constraints purge ;
drop table BPEL_SELECTORS cascade constraints purge ;
drop table BPEL_UNMATCHED cascade constraints purge ;
drop table BPEL_XML_DATA cascade constraints purge ;
drop table LARGE_DATA cascade constraints purge ;
drop table VAR_PROPERTY cascade constraints purge ;
delete from STORE_PROCESS;
```

```

delete from STORE_PROCESS_PROP;
delete from STORE_PROC_TO_PROP;
delete from STORE_VERSIONS;
delete from STORE_DU;
drop sequence hibernate_sequence;

```

1.3.3.create.sql script listing

```

-- Apache ODE - SimpleScheduler Database Schema
--
-- Apache Derby scripts by Maciej Szeffler.
--
--

CREATE TABLE ODE_JOB (
  jobid varchar2(64 char) DEFAULT '' NOT NULL,
  ts number(19,0) DEFAULT 0 NOT NULL,
  nodeid varchar2(64 char) NULL,
  scheduled number(12,0) DEFAULT 0 NOT NULL,
  transacted number(12,0) DEFAULT 0 NOT NULL,
  details BLOB, PRIMARY KEY(jobid));
CREATE INDEX IDX_ODE_JOB_TS ON ode_job(ts);
CREATE INDEX IDX_ODE_JOB_NODEID ON ode_job(nodeid);

create table BPEL_ACTIVITY_RECOVERY (ID number(19,0) not null, PIIID number(19,0), AID number(19,0), CHANNEL varchar2(255 char), REASON varchar2(255 char), DATE_TIME timestamp, LDATA_ID number(19,0), ACTIONS varchar2(255 char), RETRIES number(10,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_CORRELATION_PROP (ID number(19,0) not null, NAME varchar2(255 char), NAME_SPACE varchar2(255 char), VALUE varchar2(255 char), CORR_SET_ID number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_CORRELATION_SET (ID number(19,0) not null, VALUE varchar2(255 char), CORR_SET_NAME varchar2(255 char), SCOPE_ID number(19,0), PIIID number(19,0), PROCESS_ID number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_CORRELATOR (ID number(19,0) not null, CID varchar2(255 char), PROCESS_ID number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_CORRELATOR_MESSAGE_CKEY (ID number(19,0) not null, CKEY varchar2(255 char), CORRELATOR_MESSAGE_ID number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_EVENT (ID number(19,0) not null, IID number(19,0), PID number(19,0), TS timestamp, TYPE varchar2(255 char), DETAIL clob, LDATA_ID number(19,0), SID number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_FAULT (ID number(19,0) not null, FAULTNAME varchar2(255 char), LDATA_ID number(19,0), EXPLANATION varchar2(4000 char), LINE_NUM number(10,0), AID number(10,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_INSTANCE (ID number(19,0) not null, INSTANTIATING_CORRELATOR number(19,0), FAULT number(19,0), JACOB_STATE number(19,0), PREVIOUS_STATE number(5,0), PROCESS_ID number(19,0), STATE number(5,0), LAST_ACTIVE_DT timestamp, SEQUENCE number(19,0), FAILURE_COUNT number(10,0), FAILURE_DT timestamp, INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_MESSAGE (ID number(19,0) not null, MEX number(19,0), TYPE varchar2(255 char), DATA number(19,0), HEADER number(19,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_MESSAGE_EXCHANGE (ID number(19,0) not null, PORT_TYPE varchar2(255 char), CHANNEL_NAME varchar2(255 char), CLIENTKEY varchar2(255 char), LDATA_EPR_ID number(19,0), LDATA_CEPR_ID number(19,0), REQUEST number(19,0), RESPONSE number(19,0), INSERT_DT timestamp, OPERATION varchar2(255 char), STATE varchar2(255 char), PROCESS number(19,0), PIIID number(19,0), DIR char(1 char), PLINK_MODELID number(10,0), PATTERN varchar2(255 char)

```

```

, CORR_STATUS varchar2(255 char), FAULT_TYPE varchar2(255 char), FAULT_EXPL varchar2(255
char), CALLEE varchar2(255 char), PARTNERLINK number(19,0), PIPED_ID varchar2(255 char),
SUBSCRIBER_COUNT number(10,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primar
y key (ID));
create table BPEL_MEX_PROPS (MEX number(19,0) not null, VALUE long, NAME varchar2(255 cha
r) not null, primary key (MEX, NAME));
create table BPEL_PLINK_VAL (ID number(19,0) not null, PARTNER_LINK varchar2(100 char) no
t null, PARTNERROLE varchar2(100 char), MYROLE_EPR number(19,0), PARTNERROLE_EPR number(1
9,0), PROCESS number(19,0), SCOPE number(19,0), SVCNAME varchar2(255 char), MYROLE varcha
r2(100 char), MODELID number(10,0), MYSESSIONID varchar2(255 char), PARTNERSESSIONID varc
har2(255 char), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_PROCESS (ID number(19,0) not null, PROCID varchar2(255 char) not null u
nique, deployer varchar2(255 char), deploydate timestamp, type_name varchar2(255 char), t
ype_ns varchar2(255 char), version number(19,0), ACTIVE_ number(1,0), guid varchar2(255 c
har), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_SCOPE (ID number(19,0) not null, PIID number(19,0), PARENT_SCOPE_ID num
ber(19,0), STATE varchar2(255 char) not null, NAME varchar2(255 char) not null, MODELID n
umber(10,0), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table BPEL_SELECTORS (ID number(19,0) not null, PIID number(19,0) not null, SELGRP
ID varchar2(255 char) not null, IDX number(10,0) not null, CORRELATION_KEY varchar2(255 c
har) not null, PROC_TYPE varchar2(255 char) not null, ROUTE_POLICY varchar2(255 char), CO
RRELATOR number(19,0) not null, INSERT_TIME timestamp, MLOCK number(10,0) not null, prima
ry key (ID), unique (CORRELATION_KEY, CORRELATOR));
create table BPEL_UNMATCHED (ID number(19,0) not null, MEX number(19,0), CORRELATION_KEY
varchar2(255 char), CORRELATOR number(19,0) not null, INSERT_TIME timestamp, MLOCK number
(10,0) not null, primary key (ID));
create table BPEL_XML_DATA (ID number(19,0) not null, LDATA_ID number(19,0), NAME varchar
2(255 char) not null, SCOPE_ID number(19,0), PIID number(19,0), IS_SIMPLE_TYPE number(1,0
), INSERT_TIME timestamp, MLOCK number(10,0) not null, primary key (ID));
create table LARGE_DATA (ID number(19,0) not null, BIN_DATA blob, INSERT_TIME timestamp,
MLOCK number(10,0) not null, primary key (ID));
create table VAR_PROPERTY (ID number(19,0) not null, XML_DATA_ID number(19,0), PROP_VALUE
varchar2(255 char), PROP_NAME varchar2(255 char) not null, INSERT_TIME timestamp, MLOCK
number(10,0) not null, primary key (ID));
create index IDX_CORRELATOR_CID on BPEL_CORRELATOR (CID);
create index IDX_BPEL_COR_MESSAGE_CKEY on BPEL_CORRELATOR_MESSAGE_CKEY (CKEY);
create index IDX_SELECTOR_CORRELATOR on BPEL_SELECTORS (CORRELATOR);
create index IDX_SELECTOR_CKEY on BPEL_SELECTORS (CORRELATION_KEY);
create index IDX_SELECTOR_SELGRPID on BPEL_SELECTORS (SELGRPID);
create index IDX_UNMATCHED_CKEY on BPEL_UNMATCHED (CORRELATION_KEY);
create index IDX_UNMATCHED_CORRELATOR on BPEL_UNMATCHED (CORRELATOR);
create sequence hibernate_sequence;

```


G. Frequently asked questions

Can site consolidation utilities be used with 4th Generation Design data?

Many of the standard Teamcenter utilities are enhanced to allow their use with the large amounts of data involved in 4th Generation Design (4GD) and to support the 4GD data model. If these utilities are used in a site consolidation context, they support 4GD data as well. Additionally, the **sitcons_xfer_owner_mgr** and **sitcons_replicate_mgr** site consolidation specific utilities are enhanced to support 4GD data.

For site consolidation of unconfigured 4GD data, the export process uses the accountability, backpointer, and scratch tables to identify outdated objects. Therefore, no data traversal is required. For site consolidation of configured 4GD data, the export process uses accountability table, backpointer table, recipe, and scratch table to identify the outdated **BOMLine** objects. The process also partially expands the BOM using the **occThread** chain of outdated **BOMLine** objects and **WorksetLine** objects.

Note:

After you import 4GD data, you must run the **appmodel_fix_scope** utility at the target site to ensure that all 4GD data is represented properly. 4GD data contain partitions, you must run the **populate_top_level_partitions_cache** utility at the target site to ensure partitions are represented properly.

How are unmodifiable 4th Generation Design properties handled during import?

The import process set property values for unmodifiable properties if new object is created and its unmodifiable properties are not part of metamodel create input structure. The property is initial set to modifiable the import process to allow it to be set. After setting the value, property protection is restored.

The following are unmodifiable properties for 4th Generation Design objects.

Type name	Property name
ApprSearchCriteriaBoxZone	coordinates
ApprSearchCriteriaBoxZone	operator
ApprSearchCriteriaGroup	operator
ApprSearchCriteriaGroup	sub_criteria
ApprSearchCriteriaSavedQry	entries
ApprSearchCriteriaSavedQry	name

Type name	Property name
ApprSearchCriteriaSavedQry	values
ApprSearchCriteriaSlctState	unselected_meapprpathnodes
Cpd0ArcFilletWeld	revision_number
Cpd0ArcFilletWeld	mdl0element_thread
Cpd0ArcFilletWeld	mdl0revision_id
Cpd0ArcGrooveWeld	revision_number
Cpd0ArcGrooveWeld	mdl0element_thread
Cpd0ArcGrooveWeld	mdl0revision_id
Cpd0CollaborativeDesign	mdl0revision_id
Cpd0ConnectedElement	primary_object
Cpd0ConnectedElement	secondary_object
Cpd0ConnectedElement	user_data
Cpd0DesignElement	cpd0category
Cpd0DesignElement	cpd0id_in_parent
Cpd0DesignElement	cpd0presented_parent
Cpd0DesignElement	cpd0source_object
Cpd0DesignElement	mdl0element_thread
Cpd0DesignElement	mdl0revision_id
Cpd0DesignItemInstance	rlz0revrule_strings
Cpd0ShapeDesign	bom_view_tags
Cpd0ShapeDesign	global_alt_list
Cpd0ShapeDesign	preferred_global_alt
Cpd0ShapeDesign	revision_list
Cpd0ShapeDesign	revision_number
Cpd0ShapeDesignMaster	data_file
Cpd0ShapeDesignRevision	declared_options
Cpd0ShapeDesignRevision	gde_bvr_list
Cpd0ShapeDesignRevision	revision_number
Cpd0ShapeDesignRevision	structure_revisions
Cpd0ShapeDesignRevision	used_options
Cpd0ShapeDesignRevision	variant_expression_block
Cpd0ShapeDesignRevisionMaster	data_file
Mdl0ApprSrchSlctContent	mdl0selected_contents
Mdl0ApprSrchSlctContent	mdl0unselected_contents

Type name	Property name
Mdl0ReferenceGeometry	primary_object
Mdl0ReferenceGeometry	secondary_object
Mdl0ReferenceGeometry	user_data
Rlz0Realization	rlz0revrule_strings

What can be done if the replication process aborts because special characters are encountered during mapping?

Mapping may fail when the **sitcons_replicate_mgr** utility encounters special characters, such as illegal XML 1.0 characters, during the mapping process causing the replication process to abort. To fix this situation, create an **ILLEGAL_XML1_CHARS_SUBS** preference and set its value to a legal character or string to use in place of any illegal characters that may be encountered.

For example, to replace any illegal characters with a question mark, set the preferences value to ?:

```
ILLEGAL_XML1_CHARS_SUBS=?
```

This causes the mapping process to substitute each instance of any invalid XML 1.0 character with the specified character or string. If this preference is not set or does not have a value assigned to it, illegal XML 1.0 characters in the source database are exported as is.

What does the term site definition data include, which must be addressed during the preparation phase?

In the preparation phase, the following organizational data entities at the source site must also be defined at the target site:

User	POM_application
Group	ADA_License
Role	Tool
POM_imc	RevisionRule
NoteType	ImanVolume
DatasetType	Tc_Project
ImanType	CFM_date_info
PSViewType	TransferMode
ClosureRule	AM_ACL

IdContext	ImanActionHandler
PSOccurrenceType	ImanEventType
POM_user	GroupMember
POM_group	ResourcePool
ProjectTeam	ImanAliasList
UnitOfMeasure	

How are scalability and performance impacted by island size?

Site consolidation data transfers of large data islands can encounter scalability and performance issues. This is most likely to occur with data islands that have many **VariantExpression** and **ImanExportRecord** instances.

To avoid scalability and performance issues, place each **VariantExpression** and **ImanExportRecord** instance in its own, separate island, by setting to **TRUE** the **opt_varexp_islanchor** transfer option and the **opt_ixr_islanchor** transfer option.

- **opt_varexp_islanchor** transfer option

By default, the **opt_varexp_islanchor** transfer option is set to **TRUE**.

Siemens Digital Industries Software recommends leaving the **opt_varexp_islanchor** transfer option set to **TRUE** for site consolidation.

- **opt_ixr_islanchor** transfer option

By default, the **opt_ixr_islanchor** transfer option is set to **FALSE**. In the case where you have a large number of IXRs associated with a single island, you can set the **opt_ixr_islanchor** transfer option to **TRUE**

At the target site, the fast import reads the values of the above transfer options from the **<Header>** section of the TC XML and batches the islands with principal object of the class whose related transfer option is set to **true**. That is, if the **opt_ixr_islanchor** option is set to **true**, fast import batches the islands with the **ImanExportRecord** principal object. Similarly, if the **opt_varexp_islanchor** option is set to **true**, fast import batches islands with the **VariantExpression** principal object.

Why do we need the **fix_release_status**, **clean_up_shared_objects**, and **sitcons_fix_ixr** utilities?

These utilities address ownership inconsistencies, missing data, site administration, schema, and organizational issues.

What are closure rules?

Closure rules control the scope of the data translation on both input and output. They specify how to traverse the data structure and define relationships that are of interest and what to do when these relationships are encountered.

Closure rule benefits:

- Elegant and efficient approach.
- Codeless extensibility of data model.
- Used in other areas of Teamcenter, for example, PLM XML and Briefcase data sharing.

An example of a closure rule for item revision:

```
CLASS.ItemRevision:CLASS.Dataset:PROPERTY.IMAN_specification:TRAVERSE_AND_PROCESS
CLASS.ItemRevision:CLASS.Dataset:PROPERTY.IMAN_Rendering:TRAVERSE_AND_PROCESS
CLASS.ItemRevision:CLASS.Dataset:PROPERTY.items_tag:TRAVERSE_AND_PROCESS
CLASS.ItemRevision:CLASS.VariantExpressionBlock:PROPERTY.variant_expression_block:
TRAVERSE
CLASS.VariantExpressionBlock:CLASS.VariantExpression:PROPERTY.expressions:
TRAV+PROC
CLASS.VariantRevision:CLASS.Variant:PROPERTY.parent_variant:TRAVERSE_AND_PROCESS
CLASS.*:CLASS.icm0:PROPERTY.IMAN_classification:TRAVERSE_AND_PROCESS
CLASS.icm0:CLASS.smlb0:PROPERTY.ICUserClass:TRAVERSE_AND_PROCESS
```

What happens when a network loss or outage happens during site consolidation?

You can restart the process after a network loss or outage. It is not necessary to *back out*.

This applies to information replication, ownership change, and source site validation tables update subprocesses of site consolidation.

What can be done if replication aborts because `sitcons_replicate_mgr` mapping fails when special characters, such as illegal XML 1.0 characters, are encountered?

Create a preference as the site location, **ILLEGAL_XML1_CHARS_SUBS**, to specify substitute string characters for illegal XML 1.0 characters in the source database. For example, if the question mark character (?) is the replacement character, the preference is set as **ILLEGAL_XML1_CHARS_SUBS=?**. This replaces each instance of an invalid XML 1.0 character with the specified character or characters.

If the **ILLEGAL_XML1_CHARS_SUBS** preference is not set or does not have a value assigned to it, illegal XML 1.0 characters in the source database are exported as is.

How can I override the default delimiter in TC XML for string variable length arrays (VLAs)?

The comma character (,) is the default delimiter in TC XML for string variable length arrays (VLA).

Use the **GMS_tcxml_string_separator** preference to configure special delimiters in cases where the comma character is a legitimate character in the your data.

- The **GMS_tcxml_string_separator** preference value must be identical at the source and target sites.
- The delimiter can be multiple characters, such as **%82C**.

When you transfer ownership of data having subscription, does the subscription get transferred along with it?

There are two kinds of subscriptions; type-based and object-based. The type based ones are organizational and should be created at the administrator level, but the object-based ones should be referenced by the objects on transfer and will be carried along.

How do you determine whether unreferenced objects are orphaned?

If unreferenced objects remain after all objects and their referenced data are transferred, they are termed *orphaned* objects. Other objects or closure rules do not bind orphaned objects.

The orphaned objects are not deleted automatically. The administrator can decide whether to delete the orphaned object or transfer them.

How do you transfer objects not referenced by an item but by a folder?

Use **sitcons_replicate_mgr** with the **-folder** option to transfer a folder with its contents.

How do I configure the Teamcenter Integration Framework URL for site consolidation

For site consolidation, set the **TC_sitcons_gs_server** preference to the Teamcenter Integration Framework URL.

Can the file client cache (FCC) recognize cached data if the FMS ticket is from a different server?

The FCC caches local files and caches data by the UID of the object. Because these UIDs are preserved; the caches remain valid after consolidation is complete.

What happens to replicated files when a site is consolidated after a period of data exchange?

If the replicated files are not up-to-date, the replica objects from the target site and the corresponding files and volume data are deleted.

If the replicated files are up-to-date, the files are not deleted during consolidation. You can delete the replicated files during the cleanup phase.

Can we use volumes located near remote (former source) sites to increase operational speed?

If the volume storage of the remote site after consolidation remains the same as before consolidation, the FMS delivers files from the remote site. However, because of increased bandwidth and latency in communication, there may be performance degradation for interactions that access the Teamcenter database over a WAN.

When does the database fail?

Databases fail for many reasons such as power loss, media failure, and so on. Different database technologies have different performance and scalability characteristics. Siemens Digital Industries Software uses third-party database technologies that scale to all known Teamcenter usage parameters if they are properly administered, resourced, and maintained.

For more information on database failure modes and recovery mechanisms, see the documentation of the database software.

What is the process of deleting files from the source site if files on the target site are up to date?

Deleting files or metadata at the source site is not part of the site consolidation.

The volume handling strategy deals with files at the source site. This strategy is necessary during the execution phase since that phase potentially occurs in multiple increments and after any increment, there can still be references to the files in the source site volume storage by metadata in the source site.

The volume handling strategy during the cleanup phase will determine the deletion of the source site volume storage.

What impact do the database tuning guidelines have on the production database?

The database tuning guidelines align with standard Teamcenter database tuning guidelines. Some guidelines are general; for example, keeping statistics up-to-date. Other guidelines are a result of testing in an environment dedicated to site consolidation on a large database. Adjust these guidelines incrementally for specific databases to prevent degradation of performance.

Site consolidation tasks, like any other tasks, compete for resources and are input/output intensive. Monitor the impact on resources with tools like HP Openview and SAR. Oracle Enterprise Manager must be used by DBAs and administrators to tune and adjust deployment parameters to maintain and maximize performance and throughput.

Site consolidation utilities are memory intensive and these utilities can cause the database to encounter an out-of-memory error on systems that are low on memory. To avoid this error, your Oracle database administrator may have to turn off the Oracle Automatic Memory Management (AMM) feature to increase the available memory.

What can be done when using the `sitcons_replicate_mgr` and there is failure to connect?

Check to see whether failure to start or restart the web application servers in the right sequence. That can cause stale IP addresses in the web tier web application servers, resulting in a failure to connect. This can usually be remedied by restarting in the following sequence:

1. Start File Management System (FSC) services for the source and target sites.
2. Start pool servers for the source and target sites.

Wait for each pool server to successfully start up before continuing to the next step. You can validate the successful startups by logging into the server pool managers on the source and target site. You should see warm-started Teamcenter server entries for each environment.

3. Start the web servers for middle tiers supporting the source and target sites.
4. Start the Teamcenter Integration Framework server.

To validate that the IPS environment is ready to support data-transfer, use the following steps:

1. Use **fscadmin** on the source and target sites to verify the FSCs can ping each other.
2. Log on to the source site using the four-tier rich client.

3. Log on to the target site using the four-tier rich client.
4. Log on to the Teamcenter Integration Framework configuration interface using its URL.

When all four validation steps are successful, the data transfer launch from site consolidation orchestration should succeed.

What happens when an object's User/Group profile is not found in the target database during import?

During the site consolidation process, user and group data should be synchronized between the source and target site to avoid this situation and provide object ownership that matches what existed at the source site. However, there may be reasons for this to occur and it does not stop the import process. The owner of the object at the target site simply becomes the user used to run the import utility.

This the same behavior as in Multi-Site Collaboration when the owning user does not exist at the importing site.

How do I identify the failure point when the accountability table trigger creation fails without error information?

When you create the accountability tables you may encounter a trigger creation failure without any descriptive error displayed. To identify the problem/failure point use the following command:

```
SHOW ERROR TRIGGER trigger_name
```

Why do I get an FSC proxy error when I access a volume in a shared FMS environment?

After configuring the shared FMS environment that site consolidation requires, you may get an error message similar to the following in the rich client at the target site when you select a volume in the Organization application.

```
FSC proxy error:"FSC_Send error on getFscInfo step
./config/getfscinfo/nvargs/, lt;fsc_errort; -9409
ERROR_ADMIN_CMD_ENTERPRISEID_AND_FSCID_MISMATCH".
```

This error does not affect any Teamcenter functionality. You can ignore the message. The error may occur due to a temporary condition that exists when accessing the remote FSC after you configure the shared FMS configuration.

How do site consolidation closure rules handle incremental change elements?

When replicating an assembly using site consolidation, the incremental change (IC) elements have to be separately exported/imported by specifying the IC IDs in the input list. Although site consolidation has appropriate closure rules to identify all the relations and objects affected by an IC, these are not replicated along with the assembly and must be replicated separately. Once ICs are successfully replicated, they are automatically plugged into the replicated assembly at the target site. This does not affect performance against the closure rule based replication.

How do I resolve an import failed error that indicates the last_mod_user was not found in the database for a workspace object?

Due to the presence of legacy data or data inconsistency, a particular Teamcenter site may have objects which have a stubbed user as the last modified user (**last_mod_user** attribute) for the object. These objects fail to import on the target site during the site consolidation execution phase when you use the **sitecons_replicate_mgr** or **tcxml_import** utilities. This causes an error reported in the log file as similar to the following:

```
Import Failed with Error Xml attribute [last_mod_user = gY6knBLFgDNxsC]
not
found in database for [ImanFile].
```

This data inconsistency must be addressed during the preparation phase. For Teamcenter release 8.3.0.2 and later releases, the report generated by the **p1m_report_extract** utility displays the **last_mod_user** attribute of stubbed objects as **SC_Invalid_Stubbed_User**. You must reset the **last_mod_user** attribute of the objects to a valid user at the importing site.

The following SQL update statement resets the **last_mod_user** attribute of all the objects that have a stubbed user value as the last modified user to **tcba**:

```
update ppom_application_object set rlast_mod_useru = (select puid from
ppom_user where puser_id = 'tcdba') where rlast_mod_useru in (select
distinct
RLAST_MOD_USERU from PPOM_APPLICATION_OBJECT where RLAST_MOD_USERU NOT
IN
(Select PUID from PPOM_USER));
```

Why is my custom form object exported in a different island by itself?

When you are replicating an item or assembly through site consolidation, custom objects are not exported in the same island as the object they are associated with when you use the standard (OOTB) closure rules. The custom object may be exported in an island by itself. If you want the custom object

in same island as the object it is associated with, you must add a closure rule clause to include it in the same island.

For example, if you have a custom **Form** object attached to an **ItemRevision** object and both must be in the same island, modify the closure rule you are using to include the **Form** object. The following examples show how you modify the OOTB **siteConsolidationInternalClosureRules.xml** file to include the a custom form attached to an **ItemRevision** object with the **IMAN_specification** relation of **RELATIONSP2S** type.

Unmodified *siteConsolidationInternalClosureRules.xml* file:

```
<ClosureRule name="FT_ItemRevision"
  clauses="CLASS.ItemRevision:CLASS.Item:ATTRIBUTE.items_tag:PROCESS+TRAVERSE: :I+R,
  :
  CLASS.ItemRevision:CLASS.Form:RELATIONP2S.IMAN_master_form:PROCESS+TRAVERSE: :R,
  :
</ClosureRule>
```

Modified *siteConsolidationInternalClosureRules.xml* file

```
<ClosureRule name="FT_ItemRevision"
  clauses="CLASS.ItemRevision:CLASS.Item:ATTRIBUTE.items_tag:PROCESS+TRAVERSE: :I+R,
  :
  CLASS.ItemRevision:CLASS.Form:RELATIONP2S.IMAN_master_form:PROCESS+TRAVERSE: :R,
  CLASS.ItemRevision:CLASS.Form:RELATIONP2S.IMAN_specification:PROCESS+TRAVERSE: :R,
  :
</ClosureRule>
```

Use the following **tcxml_import** utility arguments to override the existing closure rules during import (**FT_ItemRevision** is modified in this example):

```
tcxml_import -u=tc-admin-user p=password -g=group -scope_rules
-scope_rules_mode=overwrite -file=
siteConsolidationInternalClosureRules.xml
```

Why does Teamcenter Integration Framework fail on import and display an XML/XSL file for import/export is invalid error?

You may encounter an import error message from Teamcenter Integration Framework that states:

```
XML/XSL file for import/export is invalid
```

If the export XML files show a OKB size attribute, check the available space on the drive or mount point where the export files are generated. The export operation cannot check for the sufficient disk space, resulting in an import failure when there is insufficient available space. To avoid this issue, ensure the disk that stores the export files has sufficient free space.

Why does the `sitcons_replicate_mgr` utility not submit some requests?

During a Teamcenter Integration Framework transaction that uses the `sitcons_replicate_mgr` utility with chunking, some import or export steps get stuck in progress and do not complete or show a failure in the activity status. This may occur during transactions with a large number of objects, which is usually the case when chunking is used. This is caused by the Teamcenter Integration Framework engine running out of memory. To avoid this issue:

- In the `ode-axis2.properties` file, increase the value of the `ode-axis2.db.pool.max` property, for example:

```
ode-axis2.db.pool.max=20
```

- Change the ODE to not run in memory by updating the `ode-working-dir/processes/data-transfer/deploy.xml` file as follows:
 1. Remove or comment out the `<in-memory>true</in-memory>` element.
 2. Add a `<process-events generate="none"/>` element after the `<active>true</active>` element. Delete all `.deployed` files from the directory (delete `*.deployed`) and restart the application server.

How do I transfer Audit Manager logs to the target site?

The Audit Manager log history stores standard Teamcenter events and event information in the `USERDEFINEDLOG` and `AUDITLOG` tables in the database. Site consolidation tools do not support tables that are not derived from `POM_object` based data. Therefore, you must use the utilities supplied by your database vendor to export the tables from the source site and import them into the target site. For example, for a transfer between Oracle databases at the source and target sites:

1. At the source site, enter the following database command:

```
exp db_username/db_password@SID file=backup.dmp tables=(AUDITLOG,USERDEFINEDLOG)
log=mybackup_exp.log
```

2. Transfer the `backup.dmp` file to the database location.
3. At the target site, make a backup or snapshot of your target site database and import the `backup.dmp` file into the database using the following command:

```
imp db_username/db_password@SID file=backup.dmp tables=(AUDITLOG,USERDEFINEDLOG)
log=mybackup_imp.log igonre=y
```