# TEAMCENTER

# Teamcenter Installation Using Deployment Center

Teamcenter 2412

SIEMENS

## About Siemens Digital Industries Software

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

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

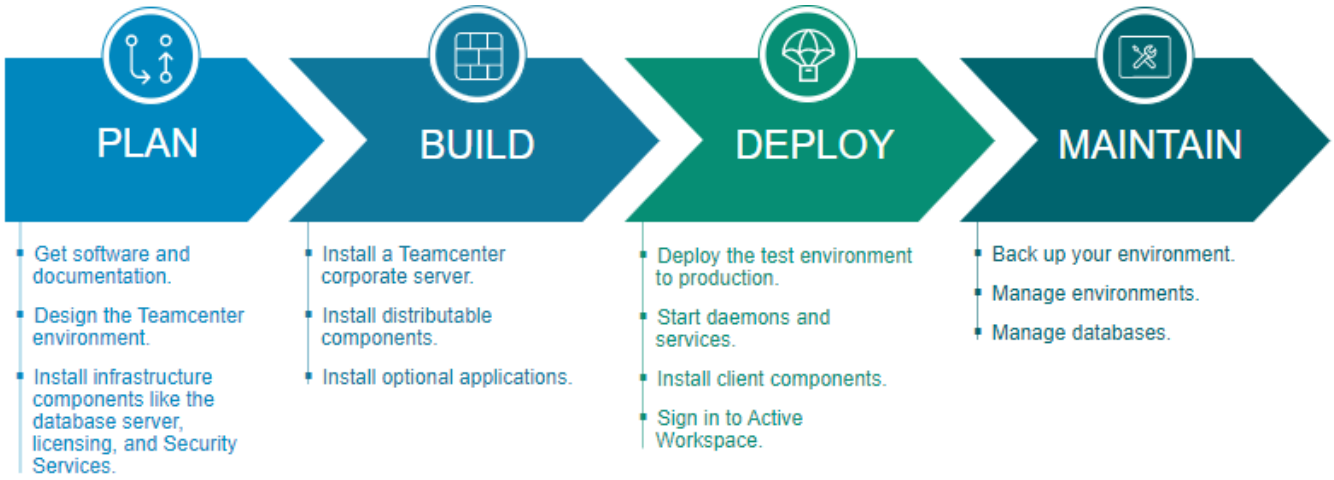
Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

# Contents

# 1. Installing Teamcenter with Active Workspace

Teamcenter is a product lifecycle management (PLM) platform that supports product development from design through manufacturing. Active Workspace is a web-based Teamcenter client with powerful collaboration, search, and visualization capabilities.

Installing Teamcenter with Active Workspace is a flexible process that accommodates the set of applications you choose from its wide selection, the geographic distribution of your users, and other variables. A Teamcenter administrator performs the installation in phases:




If you do not use Active Workspace, you can alternatively install the Teamcenter *rich client*, a Java-based desktop client available for Windows and Linux systems. Active Workspace requires *no* initial desktop installation or plug-ins like Java or ActiveX, runs in a web browser, and provides enhanced functionality compared to the rich client.

### Where do I go from here?

| If your starting point is: | And you want to: | Begin here: |
|---|---|---|
| No existing Teamcenter environment | Install Teamcenter and Active Workspace | *Plan the Teamcenter Environment* |
| Teamcenter with Active Workspace | Update Active Workspace | *Upgrade Teamcenter with Active Workspace* |
| Teamcenter without Active Workspace | Add Active Workspace | *Adding Active Workspace to a Teamcenter environment* |
| Teamcenter with rich client | Update Teamcenter and rich client | *Teamcenter Rich Client Installation on Windows* |

| If your starting point is: | And you want to: | Begin here: |
| --- | --- | --- |
| | | *Teamcenter Rich Client Installation on Linux* |

# Part I: Plan the Teamcenter Environment


Begin the Plan phase of Teamcenter installation by gathering Teamcenter documentation, software, and the Teamcenter deployment tool, Deployment Center.

Learn the architecture of a Teamcenter environment, and guidelines for distributing Teamcenter components.

The *Teamcenter Deployment Reference Architecture*, available from the Teamcenter **Downloads** area on Support Center, is an essential resource for planning a Teamcenter environment with Active Workspace. It provides information such as:

- Detailed examples of Teamcenter and Active Workspace deployments.
- Sample configurations and scripts to use with Deployment Center.
- Guidelines for test environments and development environments and how to copy a Teamcenter environment for upgrade testing.

When planning your distribution of Teamcenter components, see the Connection and Communication Table (**Teamcenter_Deployment_Connection_and_Communication_Table.xlsx**) in the *Teamcenter Deployment Reference Architecture* package. This document contains helpful guidelines for planning communication between Teamcenter components.

Prepare the machines that will host your Teamcenter test and production environments. This includes installing database infrastructure, the license server, locale support, and software like Security Services that will support the Teamcenter and Active Workspace components.

# 2. Where to start

## Get documentation

Teamcenter documentation is available from two sources:

- **Internet: Support Center**

  This is Siemens Digital Industries Software's comprehensive support portal, which provides documentation for all Siemens software products and versions.

  You require a Webkey account to access Support Center. However, you can avoid this requirement by installing the *Siemens Documentation Proxy*, which provides secure documentation access using a personalized API key, with no need to log on. Teamcenter clients can be configured to access help through the Documentation Proxy.

- **Intranet: Siemens Documentation Server**

  This is a locally installed server that can host documentation for all your Siemens Digital Industries Software products. No Internet access is required. You can configure the server for single-machine use or network-wide access with no Webkey or API key required.

  Teamcenter clients can be configured to access the help on the Siemens Documentation Server.

For an orientation to Support Center, see Siemens Software **Support Center videos** on YouTube.

### Install the Documentation Proxy or the Documentation Server

Log on to Support Center and open the **Siemens Documentation Server Downloads** page:

**Products→Siemens Documentation Server→Downloads**

Choose how you want to access documentation, and then install the Documentation Proxy or the Documentation Server.

| Installing Siemens Documentation Proxy | Installing Siemens Documentation Server |
|---|---|
| 1. Under **Select a Version**, choose **Documentation Proxy 3**, and then click the tile for the latest **Documentation Proxy 3.***x* release.<br><br>2. Download the Documentation Proxy installer:<br><br>**Windows: DocumentationProxy.***version***.exe** | 1. Under **Select a Version**, choose **Siemens Documentation Server 3**, and then click the tile for the latest **Siemens Documentation Server 3.***x* release.<br><br>2. Download the Documentation Server installer: |

| Installing Siemens Documentation Proxy | Installing Siemens Documentation Server |
| --- | --- |
| **Linux: DocumentationProxy.*version*.aol**<br><br>3. Install the Documentation Proxy according to the *Documentation Proxy Installation Guide* for Windows or Linux, available under **Release Documentation** on the software download page.<br><br>Installing the Documentation Proxy requires generating an API key at the **Siemens Support Center account site**. This may require you to obtain your Siemens site ID from your Teamcenter administrator. | **Windows: HelpServer.*version*.exe**<br><br>**Linux: HelpServer.*version*.aol**<br><br>3. Install the Documentation Proxy according to the *Siemens Documentation Server Installation Guide* for Windows or Linux, available under **Release Documentation** on the software download page. |

Note the machine and port on which you configured the Documentation Proxy or Documentation Server. These are required to configure help access from Teamcenter clients.

### Install the Teamcenter 2412 documentation kit

If you installed the Siemens Documentation Proxy, skip this section.

Teamcenter documentation is delivered in *documentation kits*. Each kit contains documentation content and an installation wizard that automatically installs documentation onto your Documentation Server.

1. Log on to Support Center and open the **Teamcenter Downloads** page:

   **Products→Teamcenter→Downloads**

2. Under **Select a Version**, choose **Teamcenter 2412**, and then click the **Teamcenter 2412** tile.

3. Install the Teamcenter 2412 documentation onto the Documentation Server:

   **Windows:** Double-click the **docs-teamcenter-2412-*locale*.exe** file.

   **Linux:** Enter the following commands:

   ```
   sudo chmod 777 docs-teamcenter-2412-locale.aol
   sudo teamcenter-2412-locale.aol
   ```

   These commands require administrative privileges.

For more information about installing documentation kits and managing the Documentation Server, see the *Siemens Documentation Server Installation Guide* for Windows or Linux.

## Verify documentation access

Open the Teamcenter 2412 documentation from your preferred source:

- Support Center (Webkey logon):

  **https://docs.sw.siemens.com/en-US/doc/282219420/PL20240523460057788.tc_doc_home**

- Support Center (via Documentation Proxy):

  **http://***doc-proxy-host***:***doc-proxy-port***/en-US/doc/282219420/PL20240523460057788.tc_doc_home**

- Siemens Documentation Server:

  **http://***doc-server-host***:***doc-server-port***/en-US/doc/282219420/PL20240523460057788.tc_doc_home**

## Enable help access in Teamcenter clients

### Configure help in the rich client

If you use the rich client, configure the **Help** button in the client to open Teamcenter help from your preferred source.

**During installation:**

When prompted in Deployment Center, enter your preferred documentation URL in the **Documentation server URL** box.

**After installation:**

Configure help access in the rich client as described in the appropriate rich client installation guide for Windows or Linux.

### Configure help in Active Workspace

Configure the **Help** button in the client to open Teamcenter help from your preferred source.

**During installation:**

The Active Workspace **Help** button links to Support Center by default and cannot be changed during installation. Accessing help directly on Support Center requires a Webkey account.

**After installation:**

If you use the Documentation Proxy or the Documentation Server, set the **TC_Help_Documentation_Link** preference to the path to your preferred documentation source. This configures the Active Workspace **Help** button to link to that source.

If you use Support Center via the Documentation Proxy, set this preference to **http://***doc-proxy-host***:***doc-proxy-port***/en-US/doc/282219420/PL20240523460057788.**

If you use Siemens Documentation Server, set the preference to **http://***doc-server-host***:***doc-server-port***/en-US/doc/282219420/PL20240523460057788.**

## Get software

Installing Teamcenter requires the Teamcenter software kit, which includes microservice framework and Active Workspace software.

1. Log on to Support Center and open the **Teamcenter Downloads** page:

   **Products→Teamcenter→Downloads**

2. Under **Select a Version**, choose **Teamcenter 2412**, and then click the **Teamcenter 2412** tile.

3. Download the Teamcenter 2412 software kit for your platform:

   - Windows: **Tc2412_wntx64.zip**
   - Linux: **Tc2412_lnx64.zip**

4. Extract its contents to a local directory.

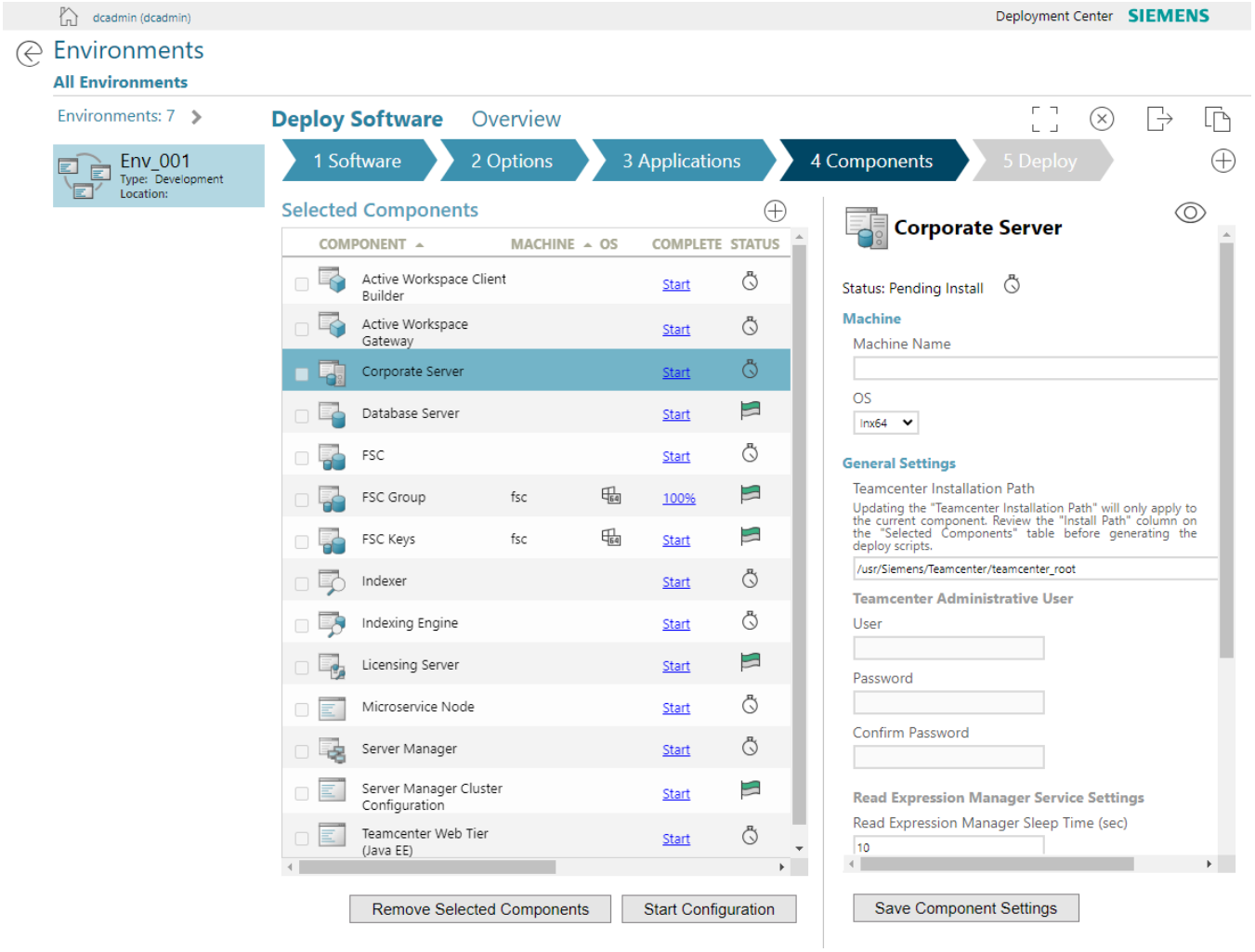
If an update (patch) to Teamcenter 2412 is available, for example, Teamcenter 2412.0001, you can additionally download the update, and apply it during the Teamcenter installation.

### Can I place the software in a remote location?

Your primary repository in Deployment Center must be a local path. However, you can specify additional repository locations, and these may be UNC paths or local file system paths. Mapped drives are not supported for any software repositories in Deployment Center. For more information, see *Deployment Center — Usage*.

## Get Deployment Center

Deployment Center is a centralized web application for deploying software to Teamcenter environments.[1] Using Deployment Center, you can create and manage multiple environments from a single location and can install and update Teamcenter software.




## Install Deployment Center

If you are familiar with Deployment Center, locate the Deployment Center software kit in the Teamcenter 2412 software kit, and then install Deployment Center 2412.

> Note:
>
> Your version of Deployment Center must be equal to or later than the version of Teamcenter you install.

If you are new to Deployment Center, prepare to install Teamcenter 2412 using Deployment Center:

1. **Learn how Deployment Center differs from TEM.**

1 Deployment Center is an alternative installation tool to Teamcenter Environment Manager (TEM) for installing Teamcenter. TEM is deprecated and will be discontinued in the next release of Teamcenter.

2. Learn how Deployment Center manages Teamcenter environments, described in *Deployment Center — Usage*.

3. Locate the Deployment Center software kit in the Teamcenter 2412 software kit as described in *Deployment Center — Usage*.

4. Install Deployment Center 2412 as described in *Deployment Center — Usage*.

### At a glance: Deployment Center vs. TEM

These two Teamcenter deployment tools use distinct approaches to building and maintaining Teamcenter environments.

- **Deployment Center**

  Deployment Center manages environments from a central machine, and generates scripts and software packages for multiple machines. Deployment Center tracks the software applications and components installed on each machine.

  In Deployment Center, selecting Teamcenter software to install primarily involves selecting *applications*, packages of administration data, software modules, and parameters that add specialized functionality to the Teamcenter environment. When you select applications, Deployment Center automatically selects the *components* required to support the selected applications. Components are the architectural pieces of Teamcenter, such as servers, services, and databases.

  You select applications in the **Applications** tab. You select and configure components in the **Components** tab.

  You can designate which machines host each component from a single instance of the Deployment Center web application. Deployment scripts supply machine information to components that communicate with each other.

- **Teamcenter Environment Manager**

  Teamcenter Environment Manager (TEM) is run on individual machines, and the Teamcenter administrator tracks what software components are installed on each machine.

  In TEM, applications and components are called *features*. Some feature groups like **Base Install** and **Server Enhancements** contain components.

  You select features (applications and components) in the **Features** panel.

  TEM refers to a collection of features that share a common Teamcenter data directory as a *configuration*. You can install multiple configurations on a single machine that share the same Teamcenter application root directory.

Run TEM on every machine where you install components. Record information about each machine to enter in configurations on other machines to enable components to communicate.

An environment created using TEM can be imported into Deployment Center by registering the environment in Deployment Center.

**Table 2-2. Comparison: Installing applications**

| Step | Deployment Center | TEM |
|---|---|---|
| 1. Select applications. | Select in the **Applications** tab. | Select in the **Features** panel. |
| 2. Select dependent components. | Selected automatically. | Select dependent components to enable features for selection. |
| 3. Enter parameter values. | Enter values in the **Components** tab. | Enter values in the sequence of panels. |
| 4. Deploy software. | Generate deploy scripts in the **Deploy** tab, and then run scripts on affected machines. | Click **Start** in the **Confirmation** panel. Repeat steps on other affected machines. |

**Table 2-3. Comparison: Installing components**

| Step | Deployment Center | TEM |
|---|---|---|
| 1. Select components. | Select in the **Components** tab. | Select in the **Features** panel. |
| 2. Enter parameter values. | Enter values in the **Components** tab. | Enter values in the sequence of panels. |
| 3. Deploy software. | Generate deploy scripts in the **Deploy** tab, and then run scripts on affected machines. | Click **Start** in the **Confirmation** panel. Repeat steps on other affected machines. |

## Get started

If you are new to Teamcenter installation, the following resources may help you get started.

| If you want to know more about: | See these resources: |
|---|---|
| Support Center | Support Center is Siemens Digital Industries Software's comprehensive support portal, providing software, documentation, and a variety of support content:<br>**https://support.sw.siemens.com**<br>For a step-by-step orientation to Support Center, see Siemens Software **Support Center videos** on YouTube. |
| Teamcenter | If you are new to Teamcenter, learn about **Teamcenter architecture and components**. |

| If you want to know more about: | See these resources: |
|---|---|
| | Also, see the *Teamcenter Deployment Reference Architecture*, which provides detailed examples of Teamcenter deployments, with sample deploy scripts and other resources. This package is available in the **Teamcenter Downloads area on Support Center**. |
| Active Workspace | If you are new to Active Workspace, learn about **Active Workspace components in Teamcenter**, and how Active Workspace installation is part of installing a Teamcenter environment. |
| Microservice Framework | Active Workspace requires the microservice framework. Learn about **microservices and the microservice framework**. |
| Deployment Center | If you are new to Deployment Center, learn how installing and managing a Teamcenter environment is different with Deployment Center. |

## System requirements

### Verify system software requirements

1. Log on to Support Center and open the **Support White Papers Certifications** page:

   a. Open **Products→Teamcenter→Downloads**.

   b. Under **Select a Version**, choose **Support White Papers→Support White Papers Certifications**, and then click the **Support White Papers Certifications** tile.

2. Download the following support documents:

   **Software Certifications Matrix (Tc2412PlatformMatrix-*date*.xlsx)**

   Contains information about system software certified for Teamcenter, such as operating systems and Java runtime environments (JREs).

   **Teamcenter Interoperability Matrix (Teamcenter Interoperability Matrix *date*.xlsx).**

   Lists versions of Siemens Digital Industries Software products that are compatible with Teamcenter 2412. It also lists supported Teamcenter paths.

   The Teamcenter Interoperability Matrix also correlates versions of Deployment Center with compatible versions of Teamcenter, and shows supported paths for upgrading Deployment Center. For information about upgrading Deployment Center, see *Deployment Center — Usage*.

Make sure you install versions of the required software that are listed in the Software Certifications Matrix and the Teamcenter Interoperability Matrix.

## Platforms

Determine from the following table which Teamcenter 2412 components are supported on your operating system. Check marks (√) indicate components supported on the given operating system.

| Operating system | Corporate server | Web tier | Active Workspace | Rich Client | Business Modeler IDE client | TCCS |
|---|---|---|---|---|---|---|
| Microsoft Windows (desktop platforms) | | | ✓ | ✓ | ✓ | ✓ |
| Microsoft Windows Server | ✓ | ✓ | | | ✓ | |
| SUSE Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Red Hat Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

### Microsoft Windows

- On Windows platforms, disable Windows User Account Control (UAC) before you install Teamcenter. This option is available in the **Control Panel→User Accounts** dialog box.

  Windows UAC can interfere with Teamcenter installation programs. Siemens Digital Industries Software recommends turning off UAC for administrative users only.

  For more information, see Microsoft Windows documentation.

- If you use a non-English language operating system version of Windows, you must install and enable the Multilingual User Interface (MUI) pack to ensure the language font is displayed properly.

  1. Download and install the MUI pack for Windows from Microsoft.
  2. Open the **Regional and Language Options** dialog box in the Windows Control Panel.
  3. In the **Languages** tab, set the required language for the menus and dialogs.
  4. In the **Advanced** tab and the **Regional Options** tab, set the required language.

### Linux

- Linux hosts must have graphics capabilities to run Teamcenter installation tools.

  For operating system requirements, see the Hardware and Software Certifications knowledge base article on Support Center.

- Linux hosts must have the **nslookup** utility available to ensure operation of the license server.

- Make sure Linux host names do not exceed 31 characters in length. Host names longer than 31 characters cause Teamcenter corporate server installation to fail during saving of the POM schema file in the *TC_DATA* directory.

  Teamcenter installation tools do not require fully qualified domain names for host names. If your fully qualified domain name exceeds 31 characters, use the server short host name instead.

  For more information, see the solutions document 002-7004480 on Support Center.

### Database

Teamcenter requires a relational database management system (RDBMS) for storing Teamcenter data. Before you install Teamcenter, you must install an Oracle database server or a Microsoft SQL Server database server.

If your database server is not a supported version, upgrade your database server to a supported version before you install Teamcenter.

Choose a database management system that suits the platforms of your Teamcenter servers and clients, and make sure your Teamcenter corporate server host has access to the database server.

If you use Oracle, set system parameters to recommended values to ensure adequate database performance.

### Web tier support

Install the required software for the Teamcenter web tier you use:

- **Java EE web tier**

  **Java Runtime Environment (JRE)**

  Install a supported JRE on the host where you build Teamcenter web applications.

  **Java EE application server**

  Install a supported application server on the host where you deploy Teamcenter web applications.

- **.NET web tier**

  **Microsoft Internet Information Server (IIS)**

  Install IIS on your Teamcenter corporate server host and add the required role services.

  **Microsoft .NET framework**

Install the .NET framework on all Teamcenter hosts.

If you use the Teamcenter Java EE web tier, install the following software:

**Java Runtime Environment (JRE)**

Install a supported JRE on the host where you build Teamcenter web applications.

**Java EE application server**

Install a supported Java EE application server on the host where you deploy Teamcenter web applications.

Some web application servers require special configuration for use with Teamcenter.

### Web browser

A web browser is required if you use the following:

- Teamcenter online help
- Active Workspace
- Deployment Center

For these products, Teamcenter supports the following web browsers:

- Windows systems: Microsoft Edge, Mozilla Firefox, and Google Chrome
- Linux systems: Mozilla Firefox and Google Chrome

For supported browser versions, see the Software Certifications Matrix on the **Support White Papers Certifications** page on Support Center.

# 3. The Teamcenter environment

## Four-tier architecture

The Teamcenter platform is a software architecture that consists of four logical *tiers* that provide the major functions:


Client interfaces through which users access Teamcenter.

Secure communication between client machines and Teamcenter servers over local, wide-area, and global networks.

Core Teamcenter operations, and retrieval and routing of Teamcenter data.

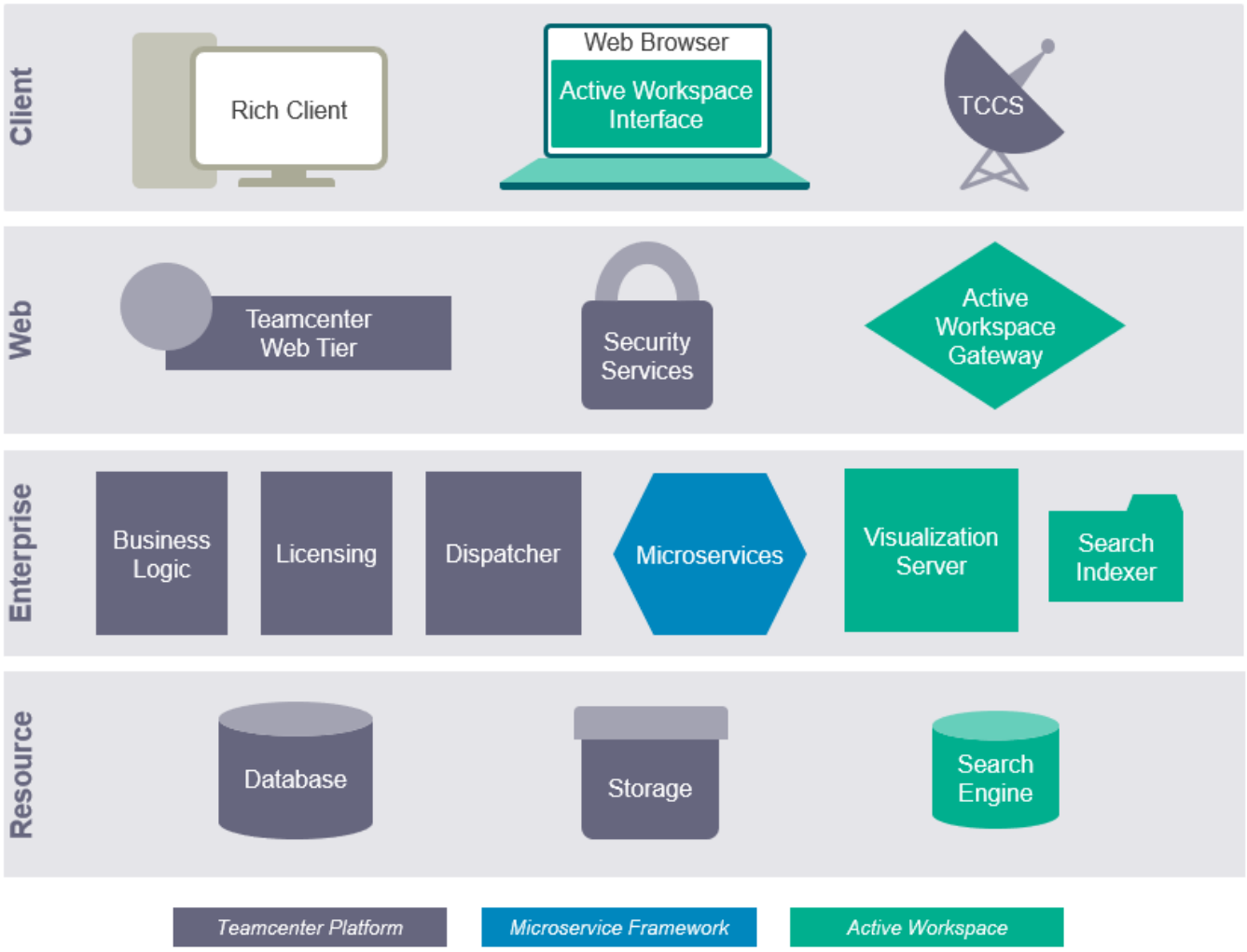
Storage and organization of user data files, metadata, and other forms of Teamcenter data.

## Teamcenter environment

Each tier of the architecture hosts Teamcenter *components*, software modules that provide supporting resources and services. Components may be installed on physical machines, virtual machines, or containers.

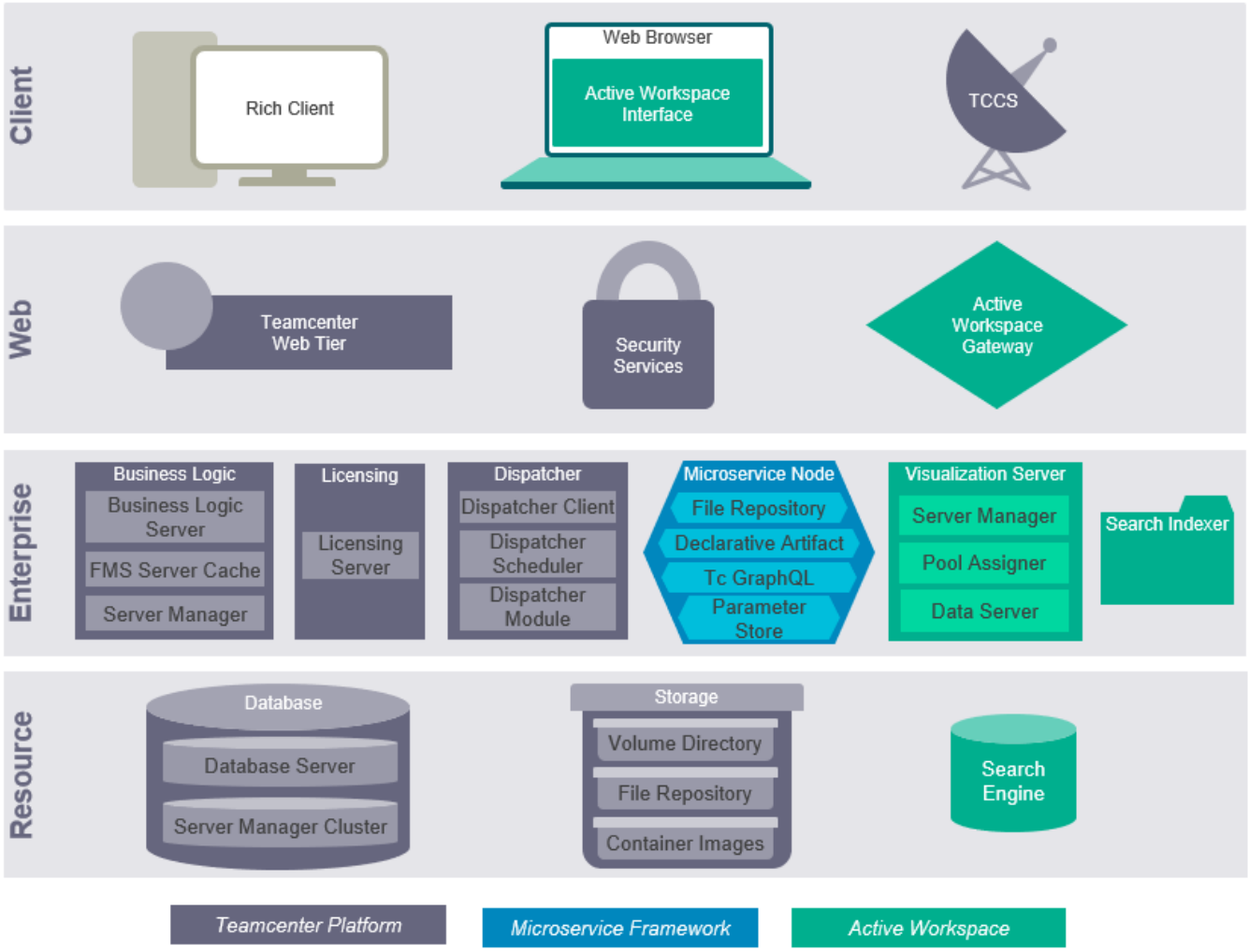
A Teamcenter *environment* consists of all client and server machines that share resources of a Teamcenter resource tier.

This simplified illustration shows groups of components representing the kinds of functionality performed in each tier.




Some components are contributed by the Teamcenter platform, some by Microservice Framework, and some by Active Workspace, as indicated.

This illustration shows names of common components in each group. These components can be selected for installation in Deployment Center:




Components can be installed on a single machine, as in a *single box* environment, or distributed on multiple machines, as in a *distributed* environment.

The *Teamcenter Deployment Reference Architecture*, available on Support Center, provides detailed examples of distributions of Teamcenter and Active Workspace components.

You can select environment types and architecture types in the **Options** tab in Deployment Center.

### Web architecture types

Teamcenter supports two third-party platforms for communication through the web tier between Teamcenter servers and clients.

**Java EE** The Java Platform, Enterprise Edition (Java EE) is supported on Windows and Linux systems. The Teamcenter **Java EE web tier** is built on the Java EE platform and requires a supported Java EE web server.

| | |
|---|---|
| **Microsoft .NET** | The Microsoft .NET framework is supported on Windows systems. The Teamcenter **.NET web tier** is built on this platform and requires Microsoft Internet Information Server (IIS). |

## Environment types

The four-tier architecture does not represent physical locations of software components, it is a logical organization for grouping components and functionality. Teamcenter components can be deployed on a single machine or multiple machines, in the following two types of environments:

| | |
|---|---|
| **Single Box** | All components are installed on one machine, and all tiers operate on that machine. This type of environment is useful for developing and testing Teamcenter deployment. |
| **Distributed** | Components are installed on multiple machines, and the functions of the four logical tiers may be distributed across multiple machines. This type is common for production environments where software functions can be distributed over a network to optimize performance with load balancing, failover support, and high availability. |

## Infrastructure types

The infrastructure type defines whether the current environment will host Teamcenter components that can be shared to multiple environments (a **Global** infrastructure) or can import components *from* other environments (a **Local** infrastructure).

| | |
|---|---|
| **Local** | Server and client components connect to the current environment. Also, client components shared from a Global infrastructure can be imported into a Local infrastructure. This is the default selection in a new environment. |
| **Global** | Components can be shared to multiple environments, and with those environments' databases. A Global infrastructure is used to define client components that can be shared to multiple environments managed in Deployment Center.<br><br>Only components that can be shared to other environments, for example, the rich client, are supported in a global infrastructure. |

# 4. Design the Teamcenter environment

## How many servers do I need?

A Teamcenter network requires one corporate server configuration. Additional servers are optional, but can help balance network loads and facilitate heterogeneous networks (networks with hosts running different operating systems).

If you install the optional servers, Siemens Digital Industries Software recommends installing in the following order:

1. Install a Teamcenter corporate server.

   The corporate server is a network node used as an application file server (from the Teamcenter application root directory) and database-specific configuration file server (from the Teamcenter data directory). Run Teamcenter Environment Manager and install the Teamcenter executables and the directory containing the database-specific configuration files. Teamcenter can also run locally on this network node.

   A Teamcenter corporate server contains the **Teamcenter Foundation** and **FMS Server Cache** features as a minimum.

2. Optionally install additional Teamcenter servers to provide the following capabilities:

   - Run Teamcenter executables and point to the existing data directory on the corporate server host or another Teamcenter server. This server can contain a Teamcenter application root directory structure on a network node that may be configured to run Teamcenter in the future.

   - Run Teamcenter Environment Manager and point to an existing database. This server can contain a Teamcenter network node to be used as a database-specific configuration file (Teamcenter data directory) server when the Teamcenter application root directory is mapped from a Teamcenter application server. Teamcenter can also be run locally on this system. You are creating an additional Teamcenter database for use with an existing Teamcenter application root directory.

### Mixed platform considerations

#### Homogeneous network environment

In a *homogeneous environment*, all hosts run the same platform, for example, a corporate server, web tier, and Teamcenter clients all running on Microsoft Windows or all running on SUSE Linux.

When deploying the two-tier architecture, you can install Teamcenter application executable files on a single application server host, export the Teamcenter application root directory structure from the Teamcenter application server, and mount it using NFS (on Linux systems) or CIFS (on Windows systems) on client workstations to run Teamcenter locally. Typically, the Teamcenter application

server is also the Teamcenter data server. Similarly, you can export the data directory structure and mount it using NFSCIFS to other Teamcenter clients to provide access to the database-specific information.

**Heterogeneous network environment**

In a *heterogeneous environment*, hosts do not all run the same platform, for example, a corporate server and a web application server may run on Linux hosts, and workstations on Microsoft Windows.

Installation considerations for a heterogeneous environment are the same as for a homogeneous environment, except that you must install Teamcenter for each type of workstation on the network, resulting in a Teamcenter application directory structure for each different type of workstation. You can configure one Teamcenter application server to serve many Teamcenter directory structures for different platforms.

Teamcenter volume data must be accessible by all Teamcenter clients in a heterogeneous network. **Configure File Management System** for volume access for all clients.

Make sure your Windows and Linux server configurations contain identical sets of Teamcenter features. For example, if you install features or custom templates on a Linux server, you must install the same features and templates on your Windows server.

Additional considerations:

- The Teamcenter root directory is platform-specific. The files within it can be shared only between systems of the same platform type. For heterogeneous Teamcenter environments that include Windows clients or Windows volume servers, configure File Management System to allow all clients to communicate with all volume servers.

- The Teamcenter root directory is specific to Windows or Linux systems (endian-specific). Maintain separate Teamcenter data directories on Windows and Linux systems.

## Planning File Management System installation

### Overview of FMS installation

File Management System (FMS) downloads and uploads file data for the rich client, embedded viewer, and Lifecycle Visualization. Multi-Site Collaboration also uses FMS servers to transfer data.

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

If the FSC does not manage any volumes, that is, if it is purely a cache server, it can run as any user that is convenient.

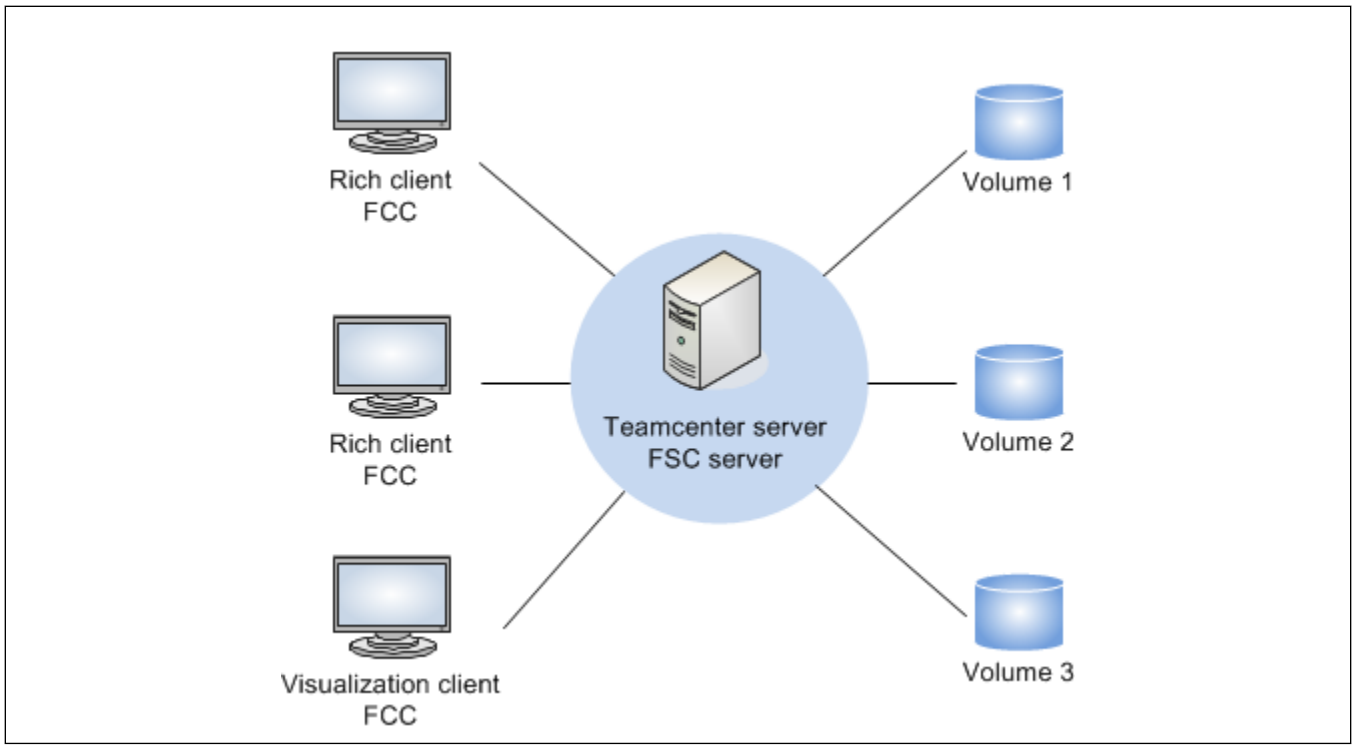
FMS provides the following functions:

- Volume server for file management
- Shared server-level performance cache for shared data access between multiple users
- Client-based private user cache for rich clients
- Transient data store mechanism for transporting reports, PLM XML, and other nonvolume data between the web and client tiers in the four-tier architecture

FMS caching enables placing the data close to the user, while maintaining a central file volume and database store.

FMS requires the installation of FMS server cache (FSC) and FMS client cache (FCC) components:

- The FSC component provides a server process and file caches for Teamcenter server hosts.
- The FCC component provides a client process and file caches for rich clients on user workstations.




**Basic File Management System deployment**

## Installing the FMS server cache

You can configure the FMS server cache (FSC) server to perform any combination of the following functions:

- Volume server or performance cache server

When running on a host where a volume is located or directly mounted on the computer hosting the FSC, the FSC acts as a volume server.

When running on a host where a volume is not located or directly mounted, the FSC acts as a performance cache server.

As a volume or cache server, the FSC checks all file access requests for a ticket that Teamcenter generates to authorize file access. As a cache server, it manages two segment caches, one for downloading files and one for uploading files.

- Configuration server

  As a configuration server, the FSC provides FMS configuration information to the FMS client caches and other FSCs.

- Transient server (in a deployment of the four-tier architecture only)

  As a transient server, the FSC delivers PLM XML and other transient files to clients.

Any deployment of Teamcenter requires a minimum of one FSC server. You can deploy multiple FSC servers, each performing multiple roles or each performing a designated purpose as either a volume, a cache, or a configuration server. When you install multiple volumes on different hosts for the same database, the multiple FSC servers are linked through a common primary (master) FSC. (You can manually configure more than one primary FSC.)

You must install an FSC server on:

- Each host running a Teamcenter server manager.
- Each host that will contain a Teamcenter volume.

FSC servers and caches are configured using XML-based files, in a hierarchical structure:

- FMS primary configuration file (**fmsmaster_***fsc_id***.xml**)

  The primary configuration file describes the File Management System network and defines FSC groups. It is the highest file in the hierarchy and can define default values for FSCs and FCCs, such as the maximum sizes of the caches.

  Each installation of Teamcenter requires one FMS primary configuration file. At least one FSC server reads this file and is called the *primary FSC*. Other FSC servers in the network download FMS configuration information from the primary FSC server.

  If you install only one FSC server in a Teamcenter network, it is the primary server.

- FSC configuration file (**fsc***fsc_id***.xml**)

The FSC configuration file configures an individual FSC in a network. It specifies the address of the primary FSC (for downloading FMS network information) and defines such values as the maximum sizes of the server segment file caches and the upload timeout value.

This file can either inherit values from the primary file or override them. It can also define default values for FCCs.

- The FCC configuration file defines values for the FCC on client hosts, such as the maximum sizes of the caches.

  It can either inherit values from the FSC configuration file or override them.

When planning your FMS installation, you must be prepared to supply the following information to the Teamcenter installation tools:

| Data | Description |
| --- | --- |
| Read cache directory and size? | For FMS to operate correctly, the location you specify must be on the local host.<br><br>If you are installing a volume on the host, FMS does not use the read cache; Siemens Digital Industries Software recommends accepting the default cache size (10 megabytes). Do not specify 0; specifying 0 creates a file cache with a default size larger than 10 megabytes.<br><br>If you are not installing a volume on this host, FMS acts as a cache server. In this case, Siemens Digital Industries Software recommends increasing the value to 1000 megabytes. However, choose a size that represents the maximum size of the data that must be processed. If you choose 1000 megabytes, and a user requests a 3 gigabyte assembly, the request fails. |
| Write cache and size? | This cache is required when the FSC acts as a cache server.<br><br>For FMS to operate correctly, the location you specify must be on the local host.<br><br>If you are installing a volume on this host, FMS does not use the write cache; Siemens Digital Industries Software recommends accepting the default cache size (10 megabytes). Do not specify 0; specifying 0 creates a file cache with a default size larger than 10 megabytes.<br><br>If you are not installing a volume on this host, FMS acts as a cache server. In this case, Siemens Digital Industries Software recommends increasing the value to 512 megabytes or more. However, choose a size that represents the maximum size of the data that must be processed. |

| Data | Description |
| --- | --- |
| Communication mode between FMS components? | Either HTTP or HTTPS. |
| Configure proxy servers? | Either HTTP proxy server or HTTPS proxy server.<br>If you choose to configure proxy servers, you must provide:<br>• The name of the host running the proxy server.<br>• The number of the port the proxy server listens on. |
| Is this host an FMS primary (master)? | If you are installing only one FSC server in the network, it must be the primary host. Each Teamcenter network must have at least one primary configuration file and one FSC designated to read this file. |
| Symmetric or asymmetric keys for ticket validation? | By default, FMS uses symmetric keys for ticket validation. You can use Deployment Center to configure your site to use more secure public-private (asymmetric) key pairs. |
| Default settings for the FCC? | • Location of the cache directory for all Windows systems and for all Linux systems.<br>• Default maximum size in megabytes of whole files downloaded from the volume to rich client hosts. Users cannot download a file whose size exceeds the value you set for this value. This default setting can be overridden by the FMS client cache configuration file.<br>Choose a size large enough to accommodate the largest whole file that users download from the volume. If the user requests a 3-gigabyte assembly when the cache size is set to 1000 megabytes, the request fails.<br>• Default maximum size in megabytes of whole files uploaded to a volume from rich client hosts. Users cannot upload a file whose size exceeds the value you set for this value. This default setting can be overridden by the FMS client cache configuration file.<br>Choose a size large enough to accommodate the largest whole file that users upload to the volume.<br>• Default maximum size in megabytes of the segment file cache used by the embedded viewer and the standalone application viewer on rich client hosts. |

| Data | Description |
| --- | --- |
| | This default setting can be overridden by the FMS client cache configuration file.<br><br>• If no or few rich client users in the network deploy Lifecycle Visualization, Siemens Digital Industries Software recommends setting this cache size to 10 megabytes. Do not specify 0; specifying 0 creates a file cache with a default size larger than 10 megabytes.<br><br>• If rich client users in the network deploy Lifecycle Visualization, Siemens Digital Industries Software recommends setting this cache size in the range of 2000 megabytes to 4000 megabytes.<br><br>The cache size is initially small, expanding to the maximum size only if a user launches Lifecycle Visualization to view a file of that size. The initial size of the cache is proportional to the value specify. |

Teamcenter installation tools install and initially configure the FSC servers, segment file caches, primary configuration file, and FSC configuration file or files. For small deployments of Teamcenter, this may be the only installation and configuration required. For large deployments, you can take advantage of FMS flexibility by manually configuring the FMS network.

## Installing the FMS client cache

The FMS client cache (FCC) process runs on a client host and performs the following functions:

- Uploads files to an FSC server
- Requests files from an FSC server
- Caches files on the client host

The FCC process manages three file caches:

- A write cache containing whole files uploaded to a Teamcenter volume
- A read cache containing whole files downloaded from a Teamcenter volume
- A segment cache for Teamcenter lifecycle visualization

Installing the FCC supports the rich client and some other Siemens Digital Industries Software products.

- The rich client requires an FCC, and Deployment Center automatically installs an FCC with each rich client.

  The rich client uploads files to the Teamcenter volume and downloads files from the Teamcenter volume using the FCC. If Teamcenter lifecycle visualization 6.0 or later is installed on the workstation and used with the rich client, it optionally uses the FCC.

  When you install the rich client on user workstations, configure the location of the cache on the workstation and the maximum size of files downloaded from the volume or uploaded to the volume. Installing the rich client on a workstation simultaneously installs the FCC process and caches. No additional configuration steps are required.

  Configuring the FCC this way may be the only configuration you require, but you can take advantage of additional configuration options by manually configuring the FCC.

- If you use NX or Teamcenter lifecycle visualization, you can install the FCC and use it to upload files to and download files from the Teamcenter volume.

  Installing the FCC enables users to take advantage of FMS features:

  - Improved file transfer performance

    FMS is a high-performance file transfer solution that gives client applications direct access to files over a high-performance network connection.

  - File streaming

    Teamcenter lifecycle visualization uses proprietary file streaming technology to download appropriate portions of the JT files over the network as they are needed. FMS supports segment file transfer to keep network loads down and support this high-performance file streaming technology.

  - Built-in caching infrastructure

    The FCC is dedicated to a specific user on the client. The FSC server can be shared by groups of users.

  - Deployment flexibility

    FMS components support a multitude of deployment configurations. This enables administrators to geographically locate volumes and shared FSC servers close to client workstations, providing the ability to tune the system for optimal file transfer performance.

  Installing an FCC for use with NX and Teamcenter lifecycle visualization is described in the Teamcenter client installation guides for Windows and Linux.

# Web tier dependencies and application integrations

## Install the web tier for four-tier rich client and Active Workspace

If you use the four-tier rich client or Active Workspace, you must install a Teamcenter web tier to provide communication between clients and the corporate server. Teamcenter provides two web tier types:

| Type | Framework | Installed using | Deployed on |
|---|---|---|---|
| .NET web tier | Microsoft .NET | Deployment Center | Microsoft Internet Information Server (IIS) |
| Java EE web tier | Java EE | Deployment Center | Any supported Java EE web server |

## Choose applications and install dependent software

Teamcenter provides many applications you can include in your environment, including integrations to third-party applications and other Siemens Digital Industries Software products. These are listed in the **Applications** tab in Deployment Center.

If you use Teamcenter integrations to other Siemens Digital Industries Software products or third-party software, install those products *before* you install Teamcenter.

Some software products require separate licenses from Siemens Digital Industries Software. Purchase the required licenses and install them into the **Siemens License Server**.

If you use integrations with the rich client, make sure you install those applications in locations specified by the Teamcenter administrator. Some of these integrations include:

- NX integrations

  Installing NX is not a prerequisite for installing or using Teamcenter, but if you intend to integrate NX with Teamcenter, install the following software before you install Teamcenter:

  - NX

    Install NX locally on every workstation according to the installation guide distributed with NX. This is required for NX integrations to function in a rich client environment.

  - Teamcenter Integration for NX or NX Integration

    Teamcenter Integration for NX and NX Integration provide the same NX user interface and are both installed with NX, but neither can be used until Teamcenter is configured.

    When you install Teamcenter Integration for NX, allow the installation to modify system files so that it can create an **installed_programs.dat** file under the **ugs** directory. You can use this

**installed_programs.dat** file as a sample on other Linux workstations of the same type to access NX and Teamcenter Integration for NX. NX can be installed on a mount point.

If you include the **NX Foundation** feature on your Corporate Server, you must install this feature on every machine where Teamcenter Foundation is installed in the environment. Also, you must install the **NX Rich Client Integration** feature on all two-tier rich clients in your environment.

When you upgrade to a new version of NX, uninstall the **NX Rich Client Integration** feature, then reinstall it, specifying the path to the new NX installation in the **NX Install Location** box in Deployment Center.

For more information about using Teamcenter with NX, see the installation guides distributed with NX.

- Teamcenter lifecycle visualization (embedded viewer)

  Download the Lifecycle Visualization software kit and install a supported version of Lifecycle Visualization on your workstation.

  When you choose this integration, Teamcenter lifecycle visualization executable files are installed on the local client host.

  Installing this feature requires system administrator privileges on the client workstation, even though the rich client does *not* require these privileges.

- Remote workflow

  When you choose this option, the rich client is enabled to support the linking of objects between Teamcenter and other applications such as Teamcenter portfolio, program and project management. Separate installation of remote workflow components are required.

# 5. Configure language support

## Supported Teamcenter localizations

Siemens Digital Industries Software provides localized versions of Teamcenter in the following locales:

| Language | Locale code |
|---|---|
| Chinese (Simplified) | **zh_CN** |
| Chinese (Traditional) | **zh_TW** |
| Czech | **cs_CZ** |
| English | **en_US** |
| French | **fr_FR** |
| German | **de_DE** |
| Italian | **it_IT** |
| Japanese | **ja_JP** |
| Korean | **ko_KR** |
| Polish | **pl_PL** |
| Portuguese (Brazilian) | **pt_BR** |
| Russian | **ru_RU** |
| Spanish | **es_ES** |

Use the appropriate locale codes to deploy Teamcenter localizations or launch Teamcenter clients in a desired locale.

If you provide your own localizations for locales not provided by Siemens Digital Industries Software, use the appropriate Java standard locale codes similar to the locale codes in the preceding table.[1]

### Localizing Teamcenter in Hebrew

Siemens Digital Industries Software does not provide a Hebrew translation but provides recommended configuration settings for Hebrew locales. In Hebrew locales, set the locale code to **en_US**. This allows data entry in Hebrew, but user interface text is in English.

1 Standard locale codes are composed of a two lowercase character language code from the ISO 639-1 standard, followed by an underscore, followed by a two uppercase character country code from the ISO 3166-1-alpha-2 standard.

## Choose the character set for Teamcenter

Choosing the correct character set for Teamcenter and the Teamcenter database is critical. If a Teamcenter client user enters a character that is not recognized by the Teamcenter database, the character is misinterpreted or corrupted when the user's data is checked into the Teamcenter database.

Determine the character set your Teamcenter network requires based on the following considerations.

### Language support

Determine the languages you need to support, considering both initial needs and future needs. If you support one language currently but anticipate supporting additional languages in the future, choose a character set that accommodates those future requirements.

Some character sets support groups of languages. The **standard localizations provided with Teamcenter** support the following language groups:

| Language group | Languages |
|---|---|
| Western European | English<br>French<br>German<br>Italian<br>Portuguese (Brazilian)<br>Spanish |
| Eastern European | Czech<br>Polish<br>English |
| Japanese | Japanese<br>English |
| Chinese (Simplified) | Chinese (Simplified)<br>English |
| Chinese (Traditional) | Chinese (Traditional)<br>English |
| Korean | Korean<br>English |
| Russian | Russian<br>English |

If the languages you plan to support are all in the same language group, you may choose a non-UTF-8 character set for your Teamcenter network. But, if you plan to support languages that are *not* all within a single language group, you must choose the UTF-8 character set.

For example, if your Teamcenter hosts run in English, French, and German locales, which are all in the Western European language group, you may choose a non-UTF-8 character set *or* you may choose UTF-8. However, if you also need to support hosts in Japanese locales, you must choose UTF-8 because Japanese is not in the Western European language group.

The UTF-8 character set supports *all* languages supported by standard Teamcenter.

### Choosing UTF-8 or non-UTF-8

Unicode encodings like UTF-8 enable seamless manipulation of all existing characters in all languages. Teamcenter supports non-Unicode and UTF-8 Unicode encodings.

In a system fully configured for UTF-8 (for example, a server host configured for UTF-8 and a database encoding of Oracle **utf8** or Oracle **al32utf8**), all characters can be entered in the application.

In a system configured for a non-Unicode encoding, only characters belonging to it can be entered. ASCII characters are always part of that character list. For example, if you choose Western European setup (Microsoft **cp1252** or ISO **iso-88559-1** encodings), you cannot enter Russian, Japanese, Chinese, Czech, Polish, Taiwanese, or Korean characters. Furthermore, database migration from one encoding to Unicode can be tedious. It is important, then, to fully consider present and future needs when choosing encoding.

### Character support

Determine what special or extended characters you must support in Teamcenter data, and choose a character set that supports them. For example:

**En dash (–) or em dash (—)**

These characters are part of Windows 1252 code page, but not part of the **ISO8859_1** character set. However, the **UTF-8** character set supports these characters.

**Currency symbols such as the euro (€)**

This symbol is in the **we8iso8859p15** character set, but not in the **we8iso8859p1** character set.

To ensure correct character mapping, make sure the database and the Teamcenter server use the same encoding.

### Platform and database

- **Platform**

  Choose a character set that accommodates the platforms in your Teamcenter network. For example, if your Teamcenter server is a Linux host but your client hosts are Windows, and you use default character sets on each, data corruption can result because the default character sets for these platforms are not compatible. Choose a character set supported on both platforms.

The UTF-8 character set accommodates all platforms Teamcenter supports.

- **Database**

  Oracle supports UTF-8 and non-UTF-8 character sets on all platforms.

  Microsoft SQL Server does not provide native support for UTF-8. However, you can configure Teamcenter to use UTF-8 with a Microsoft SQL Server database. The **Enable UTF-8 Mode?** parameter in the **Database Server** component in Deployment Center enables the Teamcenter server to convert character encoding to and from UTF-8 when interacting with the database.

## Verify that your locale is supported

On Windows systems, if you do not use UTF-8, ensure the locale you want to use is supported on your host. Perform the following steps to set the Windows system locale and install the required language packs:

1. Open the **Regional and Language Options** dialog box in the Windows Control Panel.
2. In the **Languages** tab, set the required language for the menus and dialog boxes.
3. In the **Region and Language** dialog box, click the **Administrative** tab.
4. Under **Language for non-Unicode programs**, click **Change system locale**.
5. In the **Region and Language Settings** dialog box, verify the correct locale (language and country) is selected. If not, choose the correct locale.
6. Close all dialog boxes and restart your system to install and configure the required language pack.

On Linux systems, to verify that the desired locales are supported on your host, type the following command:

```
locale -a
```

This command returns a list of all locales the host supports. If a locale you need is not included in the list, contact your system administrator to install the required language pack.

Keep in mind that some Linux platform GUIs may allow you to set a locale that is not in the list of supported locales on the host. Make sure the locale you set is supported on the host.

To verify that a desired character set is available on your host, type the following command, which lists character sets supported on the host:

```
locale charmap
```

# Configuring a UTF-8 environment for Teamcenter

## Overview of UTF-8 configuration

Teamcenter supports the Unicode UTF-8 character set on Windows and Linux hosts that are configured to process UTF-8.

To configure your Teamcenter host to use Unicode UTF-8, perform the following steps before you install Teamcenter:

1. Configure your operating system to run Unicode UTF-8.

2. Install a database server and enable Unicode UTF-8 character set support during installation.

To configure your Teamcenter host to use the UTF-8 character set, install a database server and enable UTF-8 character set support during installation.

**Set the required values for your platform, locale, and database type** *before* you begin installing Teamcenter.

**Enable UTF-8 support for Teamcenter servers and clients during Teamcenter installation**

- **Teamcenter servers**

  With UTF-8 support configured on your host, the deploy script from Deployment Center can create a UTF-8-enabled Teamcenter database during Teamcenter installation.

  If you use Microsoft SQL Server, select the **Enable UTF-8 Mode?** option in the **Database Server** component in Deployment Center.

- **Two-tier rich client**

  You can select **UTF8** encoding in the **Character Encoding Settings** parameter in the **Database Server** component in Deployment Center.

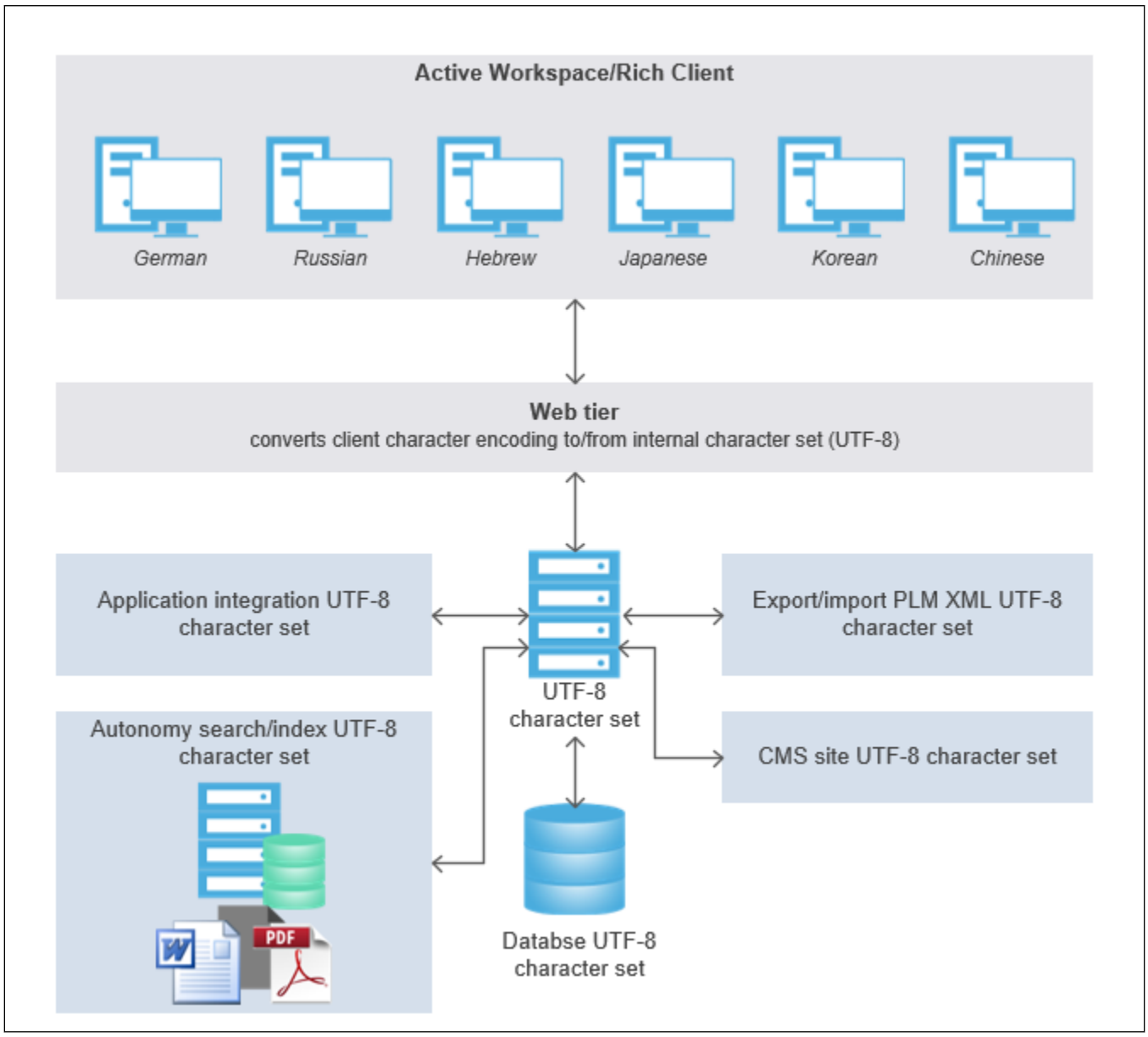
- **Web tier**

  Make sure UTF-8 support is configured on the web tier host.

  The web tier can run on any Windows or Linux platform running any language character set. The Teamcenter web tier converts client character encoding to and from UTF-8 as it passes through the web tier.

The following example shows a Teamcenter configuration for restricted Unicode UTF-8 character set support with clients displaying multiple locales. Servers in this configuration run a Unicode UTF-8 character set operating system.

On Windows platforms, if the database is configured for the UTF-8 character set, the Teamcenter server operates in UTF-8 mode independent of the system locale.




**Unicode homogeneous server platform configuration**

- Your Linux platform administrator must configure the host to run the Unicode UTF-8 character set operating system by default. This enables all software running on in the operating system environment to recognize the default character set is UTF-8.

- Teamcenter does not support Unicode Supplementary Characters.[2]

2 Unicode Supplementary Characters are characters in the Unicode Character Standard outside of the Basic Multilingual Plane (BMP), that is, characters with code point values larger than 0xFFFF.

- If you import translated content in languages that require multibyte characters, such as Russian and Chinese Simplified, you must configure your Teamcenter installation to support the UTF-8 character set to ensure that titles and other properties display correctly in your environment.

## Configure UTF-8 environment settings

If you use UTF-8, select the **al32utf8** or **utf8** character set when you install your database server.[3]

For Microsoft SQL Server, no special setting is needed during database server installation. If you select the **Enable UTF-8 Mode?** option in the **Database Server** component in Deployment Center, the Teamcenter server converts character encoding to and from UTF-8. This allows Teamcenter to use UTF-8 with Microsoft SQL Server's (non-UTF-8) internal encoding.[4]

In addition, on Linux systems, set the **LANG** and **LC_ALL** system environment variables to the appropriate values for your locale and platform. These variables must have identical values to function properly.

**Values for LANG and LC_ALL**

| Locale | Value |
|---|---|
| Chinese (Simplified) | **zh_CN.utf8** |
| Chinese (Traditional) | **zh_TW.utf8** |
| Czech | **cs_CZ.utf8** |
| English | **en_US.utf8** |
| French | **fr_FR.utf8** |
| German | **de_DE.utf8** |
| Hebrew | **he_IL.utf8** |
| Italian | **it_IT.utf8** |
| Japanese | **ja_JP.utf8** |
| Korean | **ko_KR.utf8** |
| Polish | **pl_PL.utf8** |
| Portuguese (Brazilian) | **pt_BR.utf8** |
| Russian | **ru_RU.utf8** |
| Spanish | **es_ES.utf8** |

In Hebrew locales, set the following additional variables:

1. In the *TC_DATA*/**tc_profilevars** file, set **TC_XML_ENCODING** to **UTF-8**.

3 Oracle recommends **al32utf8**. **UTF8** supports only supports Unicode Version 3.0 and earlier.
4 Microsoft SQL Server does not provide native support for UTF-8 character set encoding.

2. In two-tier environments, set **TC_CHARACTER_ENCODING_SET** to **UTF8** in the following files:

   - *TC_ROOT*/**tccs/Start_TcServer1**
   - *TC_ROOT*/**pool_manager/confs/MYDB/mgrstart**

Do not set the **TC_XML_ENCODING** or **TC_CHARACTER_ENCODING_SET** environment variables in the system environment. TEM sets these values in the Teamcenter configuration.

## Configuring a non-UTF-8 environment for Teamcenter

To ensure correct display and processing of Teamcenter data, set the required values in your operating system environment. Use the appropriate values for your locale and platform.

### Environment settings on non-UTF-8 systems

| | | Value | |
|---|---|---|---|
| **Locale** | **Setting** | **Linux** | **Microsoft Windows** |
| Chinese (Simplified), GB2312-80 encoding | Database character set (Oracle) | **zhs16cgb231280** or **zhs16gbk** | **zhs16cgb231280** or **zhs16gbk** |
| | Database collation (MS SQL Server)[1] | N/A | **chinese_prc_bin** |
| | **LANG** and **LC_ALL**[2] | **zh_CN** | N/A |
| Chinese (Simplified), GBK encoding | Database character set (Oracle) | **zhs16cgb231280** or **zhs16gbk** | **zhs16cgb231280** or **zhs16gbk** |
| | Database collation (MS SQL Server)[1] | N/A | **chinese_prc_bin** |
| | **LANG** and **LC_ALL**[2] | **zh_CN.gb18030** | N/A |
| Chinese (Traditional) | Database character set (Oracle) | **zht16big5** or **zht16mswin950** | **zht16big5** or **zht16mswin950** |
| | Database collation (MS SQL Server)[1] | N/A | **chinese_taiwan_stroke_bin** |
| | **LANG** and **LC_ALL**[2] | **zh_TW** | N/A |
| Czech | Database character set (Oracle) | **ee8mswin1250** | **ee8mswin1250** |
| | Database collation (MS SQL Server)[1] | N/A | **czech_bin** |
| | **LANG** and **LC_ALL**[2] | **cs_CZ** | N/A |

**Notes:**

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

| | | Value | |
|---|---|---|---|
| **Locale** | **Setting** | **Linux** | **Microsoft Windows** |
| English | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **en_US** or **en_US.iso885915** | N/A |
| French | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **fr_FR**[5] | N/A |
| German | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **de_DE**[5] | N/A |
| Hebrew[6] | Database character set (Oracle) | **iw8iso8859p8** or **iw8mswin1255** | **iw8iso8859p8** or **iw8mswin1255** |
| | Database collation (MS SQL Server)[1] | N/A | **hebrew_bin** |
| | **LANG** and **LC_ALL**[2] | **iw_IL.utf8** | N/A |
| Italian | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **it_IT**[5] | N/A |
| Japanese (EUC) | Database character set (Oracle) | **ja16euc** | **ja16euc** |
| | Database collation (MS SQL Server)[1] | N/A | N/A |
| | **LANG** and **LC_ALL**[2] | **ja_JP.eucjp** | N/A |

**Notes:**

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

| | | Value | |
|---|---|---|---|
| **Locale** | **Setting** | **Linux** | **Microsoft Windows** |
| Japanese (Shift-JIS) | Database character set (Oracle) | **ja16sjis** | **ja16sjis** |
| | Database collation (MS SQL Server)[1] | N/A | **japanese_bin** |
| | **LANG** and **LC_ALL**[2] | **ja_JP.sjis** | N/A |
| Korean | Database character set (Oracle) | **ko16ksc5601**[7] | **ko16ksc5601**[7] |
| | Database collation (MS SQL Server)[1] | N/A | **korean_wansung_bin** |
| | **LANG** and **LC_ALL**[2] | **ko_KR.EUC** | N/A |
| Polish | Database character set (Oracle) | **ee8mswin1250** | **ee8mswin1250** |
| | Database collation (MS SQL Server)[1] | N/A | **polish_bin** |
| | **LANG** and **LC_ALL**[2] | **pl_PL.ISO8859-2** | N/A |
| Portuguese (Brazilian) | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **pt_BR**[5] | N/A |
| Russian | Database character set (Oracle) | **cl8mswin1251** or **cl8iso8859p5** | **cl8mswin1251** or **cl8iso8859p5** |
| | Database collation (MS SQL Server)[1] | N/A | **cyrillic_general_bin** |
| | **LANG** and **LC_ALL**[2] | **ru_RU** | N/A |
| Spanish | Database character set (Oracle) | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] | **we8iso8859p1** or **we8iso8859p15**[3] or **we8mswin1252**[4] |
| | Database collation (MS SQL Server)[1] | N/A | **latin1_general_bin** |
| | **LANG** and **LC_ALL**[2] | **es_ES**[5] | N/A |

**Notes:**

1. The database collation you select during Microsoft SQL Server installation determines the database character set.
2. Set **LANG** and **LC_ALL** in the system environment variables. These variables must have identical values to function properly.
3. **we8iso8859p15** contains additional characters, including the euro symbol (€).
4. **we8mswin1252** contains more characters than **ISO-8859-15**.
5. No **ISO-8859-15** equivalent is available for this locale.
6. Siemens Digital Industries Software does not provide a Hebrew translation. The configuration settings shown allow data entry in Hebrew, but user interface text is in English.
7. If you migrate to **ko16ksc5601** from UTF-8, some data may be truncated. You must modify truncated valued because Teamcenter does not support modifying the default field size.

In Hebrew locales, set the following additional variables:

1. In the *TC_DATA*/**tc_profilevars** file, set **TC_XML_ENCODING** to **ISO-8859-8**.

2. In two-tier environments, set **TC_CHARACTER_ENCODING_SET** to **ISO8859_8** in the following files:

   - *TC_ROOT*/**tccs/Start_TcServer1**
   - *TC_ROOT*/**pool_manager/mgrstartMYDB**

Do not set the **TC_XML_ENCODING** or **TC_CHARACTER_ENCODING_SET** environment variables in the system environment. The deploy script sets these values in the Teamcenter configuration.

For non-English locales on Linux systems, you must specify the system locale when logging on to the system using KDE.

## Verify required character set

You must have the same locale installed on your Teamcenter host as you use to communicate with your database server, and the database server must support this locale as well.

On Linux systems, Teamcenter installation tools, verify that the required character set is loaded by running the **locale -a** command in a shell. If the output does not list the required character set, you must add this character set before you install Teamcenter.

1. Set or export the **LC_ALL** environment variable by typing **LC_ALL=***character-set* or the equivalent command for your platform.

2. Verify the setting using the **echo** command or equivalent. Make sure the correct value for **LC_ALL** is displayed.

3. Run the **locale** command and make sure the **LANG** variable and all the **LC_***x* variables are set the same as **LC_ALL**.

4. If **LANG** is still set to **C**, manually export **LANG** to be the same value as **LC_ALL**.

5. Launch Teamcenter Environment Manager (**tem.sh**) from the current shell.

Alternatively, your system administrator may modify the date file (named **TIMEZONE** in the **etc** directory), which can preset this environment, so every time you log on and launch a shell, the environment is preset.

The recommended method, however, is to log on to the system using the Common Desktop Environment (CDE) with the minimum required locale by choosing **Option→Language→***character-set* during logon.

If the required character set is not loaded on your machine, contact your system administrator to have it installed before you install the GM Overlay.

This requirement is necessary because current Teamcenter versions use XML files rather than **.dat** files and associated scripts. Because of this, GM Overlay data is transformed from **.dat** files into XML files.

To read and parse the XML files correctly, the system must be able to process non-English (non-ASCII) locale characters. To facilitate this, the system must be first loaded with the fonts for that locale.

## Choose the default language for the Teamcenter server process

Teamcenter server (TcServer) processes and other Teamcenter processes, and Teamcenter command-line utilities, start in the language specified in the **TC_language_default** environment variable. To make these display in a different preferred locale, set the **TC_language_default** environment variable to a supported locale codea supported locale code.

Teamcenter allows users to select a locale on their client hosts, regardless of the locale used by the Teamcenter server pool manager. Requested locales *must* be installed on the Teamcenter server (which may not be true for customized locales) and the server system be configured to accept the locale encoding.

# 6. Installing a database server

## Install a database server

Teamcenter requires a supported relational database management system (RDBMS) for storing Teamcenter data. Before you begin installing Teamcenter, you must install and configure one of the following supported database systems:


- Oracle
- Microsoft SQL Server

Before proceeding with database server installation, make sure you are correctly licensed through your database vendor for the database edition you install.

For information about database versions supported for use with Teamcenter, see Support Center.

Because of Teamcenter's high resource demands, Siemens Digital Industries Software recommends a dedicated database server. At a minimum, there should be a dedicated database instance for Teamcenter. This allows the instance to be tuned specifically for Teamcenter.

## Install and configure Oracle

### Preparing the Oracle server

Your Oracle database server must be a version certified for use with Teamcenter 2412. For information about certified Oracle versions, Oracle disk space requirements, and operating system and service patch requirements, see the Hardware and Software Certifications knowledge base article on Support Center.

You may choose to create a new Oracle database or upgrade existing Oracle databases. Install a certified version of Oracle Server if a certified version is not installed on the system. For certified Oracle versions and disk space requirements, see the Hardware and Software Certifications knowledge base article on Support Center.

Teamcenter supports pluggable databases (PDB) with container databases (CDB) if you use Oracle 12c or later.

When installing the database server:

- Choose as an Oracle database server a host that is directly accessible by the Teamcenter server host. A database server host is usually a dedicated high-capacity server, specifically tuned for Oracle.
- Install Oracle on each database server or NFS-mount Oracle to each database server.

- Create databases locally on servers.

You can install Oracle from either of the following sources:

- Oracle software kit supplied by Siemens Digital Industries Software
- Oracle software kit supplied by Oracle Corporation

Prepare an Oracle database server and configure an Oracle database for Teamcenter:

1. Choose a name for the Oracle user for the Teamcenter database. Teamcenter uses this account as the owner of all Teamcenter-created tables. This account is used by the database administrator to perform tasks required by Teamcenter.

2. On Linux systems, **set shell limits and parameters on the Oracle server host**..

3. If you do not have a certified version of Oracle, install or upgrade Oracle:

   - If you do not have an Oracle server installed, **install a certified version of Oracle**.

   - If you have an Oracle server installed, but it is not a version certified for Teamcenter 2412, **upgrade your Oracle server**.

4. **Configure Oracle software** for Teamcenter.

5. **Create a database for Teamcenter**.

To ensure correct character mapping, make sure the database and the Teamcenter server use the same encoding.

### Additional database instances

Create a database instance if one does not exist or if an additional database instance is required, for example, to support testing, training, or Repeatable Digital Validation (RDV).

If you are installing Repeatable Digital Validation (RDV) services, Siemens Digital Industries Software recommends strongly that you create a *new* database instance on an Oracle server with database partitions on a separate drive. RDV requires extensive data warehousing with large uploads and simple queries. Such a configuration also makes the fine-tuning of the database easier.

A separate RDV database is *not* required if you use cacheless search.

## Set shell limits and parameters

### Overview of shell limits and parameters

Oracle RDBMS uses extensive Linux resources such as shared memory, swap memory, and semaphore for interprocess communication. Inadequate parameter settings cause problems during installation and startup. Increasing the volume of data stored in memory reduces disk I/O activity and improves database performance.

The Oracle RDBMS installation program displays warnings if kernel parameters are not adequate. To avoid warnings and errors during or after installation, make sure kernel parameters meet the recommended settings for typical environments described in the following topics.

Before you install Oracle RDBMS, set initial parameters as described in Oracle documentation, and then adjust parameters according to available system memory. Set the **ulimit** parameter to **unlimited**.[1] Then, set the **kernel parameters** to recommended Teamcenter values for your operating system.

If you previously tuned kernel parameters for other installed applications to levels that meet or exceed the values recommended for Teamcenter, keep those existing values.

The parameter settings recommended herein are *minimum* values. For production database systems, Oracle recommends you tune values to optimize system performance. For information about performance tuning, see:

- Documentation for your operating system
- Teamcenter installation documentation on Support Center

### Set SUSE Linux shell limits

1. Increase shell limits for the **oracle** user to the minimum values listed in the following table by adding the following lines to the **/etc/security/limits.conf** file:

```
oracle              soft    nproc   2047
oracle              hard    nproc   16384
oracle              soft    nofile  1024
oracle              hard    nofile  65536
```

   Do not change the shell limit values if they were set for another program and the values are greater than the levels Oracle requires.

1 The **ulimit** parameter specifies a maximum number of processes per user.

| SUSE Linux shell limit | Item in limits.conf | Minimum hard limit |
|---|---|---|
| Maximum number of open file descriptors | **nofile** | 65536 |
| Maximum number of processes available to a single user | **nproc** | 16384 |

2. Add or edit the following lines in the **/etc/pam.d/login** file:

```
session required /lib64/security/pam_limits.so
session required pam_limits.so
```

3. Change the **oracle** user default shell startup file:

   - For the Bourne, Bash, or Korn shell, add the following lines to the **/etc/profile.local** file:

```
if [ $USER = "oracle" ]; then
        if [ $SHELL = "/bin/ksh" ]; then
              ulimit -u 16384
              ulimit -n 65536
        else
              ulimit -u 16384 -n 65536
        fi
fi
```

   - For the C shell (csh or tcsh), add the following lines to the **/etc/csh.login.local** file:

```
if ( $USER == "oracle" ) then
        limit maxproc 16384
        limit descriptors 65536
endif
```

## Upgrade an Oracle server and database

### Export an Oracle database

Windows systems:

1. Log on to the Oracle server as an administrator user.

2. Export the contents of your Teamcenter Oracle database to the dump file:

```
ORACLE_HOME\bin\expdp db-user/password full=y dumpfile=file-name.dmp
logfile=export.log
```

Replace *db-user* with the Teamcenter database user account name; replace *password* with the database user account password; replace *file-name* with the full path and name of the dump file to contain the exported data; replace *export* with the name of the log file to contain export output.

3. Store the dump file in a safe place.

Linux systems:

1. Either log on to the Oracle server as **oracle** or switch the user to **oracle**:

   ```
   su - oracle
   ```

2. Set the **PATH** environment variable to include the Oracle **bin** directory:

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

3. Manually set the shared library path for Linux:

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

4. Export the contents of the Teamcenter Oracle database to the dump file:

   ```
   ORACLE_HOME/bin/exp db-user/password full=y file=file-name.dmp
   log=export.log
   ```

   Replace *db-user* with the Teamcenter database user account name; replace *password* with the database user account password; replace *file-name* with the name of the dump file to contain the exported data; replace *export* with the name of the log file to contain export output.

5. Store the dump file in a safe place.

If you have multiple databases, repeat this procedure for each database.

> Caution:
>
> Siemens Digital Industries Software strongly recommends backing up the dump file on tape or another disk. If the dump file becomes corrupted or lost, all data from the existing database is lost.

### Terminate Oracle sessions on Windows systems

#### Stop the listener process

1. Log on to the operating system as a user with administrator privileges.

2. Open the **Services** dialog box in the Windows Control Panel.

3. Select the Oracle TNS listener services (**Oracle***release-ID***TNSListener**) and click **Stop**.

#### Shut down an Oracle database

### Shut down Oracle using Windows Control Panel

1. Log on to the operating system as a user with administrator privileges.
2. Open the **Services** dialog box in the Windows Control Panel.

   Windows displays the Services window.
3. Select the **OracleService***SID* service.

   Replace *SID* with the system identifier of the database instance.
4. Click **Stop**.

### Shut down Oracle using SQL*Plus

1. Log on to the operating system as a user with administrator privileges.
2. Start the Oracle SQL*Plus utility:

   ```
   sqlplus sys/password@Oracle-SID as sysdba
   ```

   Replace *password* with the password for the **sys** user account.

   Oracle starts the Oracle SQL*Plus utility.

   The **sys** user must be in the Oracle **sysdba** group for the Oracle system identifier (SID) used by Teamcenter. To connect as internal (without a password), the account must be part of the **ORA_DBA** local group in Windows.
3. Shut down the database instance by typing the following command:

   ```
   shutdown
   ```
4. Exit SQL*Plus:

   ```
   exit
   ```

#### Terminate Oracle sessions on Linux systems

Before installing a new version of Oracle, you must terminate all Oracle sessions and Oracle processes.

1. Either log on to the Oracle server as **oracle** or switch the user to **oracle** as follows:

   ```
   su - oracle
   ```

2. Set the **ORACLE_HOME** environment variable to point to the location of the Oracle files. For example:

   ```
   export ORACLE_HOME=/u01/app/oracle/product/oracle-version
   ```

   Replace the path with the system path to the Oracle files.

3. Define **ORACLE_HOME/bin** in the **PATH** variable:

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

4. Manually set the shared library path or Linux:

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

5. If a **tnslsnr** listener process is running, terminate it. For example:

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

   Replace *listener-name* with the name of the listener process.

6. Shut down all Oracle database instances using the **dbshut** utility. Shut down database instances listed in the **oratab** file:

   ```
   $ORACLE_HOME/bin/dbshut
   ```

### Back up an Oracle installation

If you are upgrading to the certified Oracle version, back up the existing Oracle installation.

Backing up your Oracle installation before upgrading is strongly recommended. Failure to back up existing data could result in loss of data if problems occur during the upgrade process.

Back up the following files and directories:

- The Oracle home directory on each installed workstation.
- The directories containing database files for each configured database.
- The Oracle Net **listener.ora** and **tnsnames.ora** configuration filesin the **/etc** directory.

These are the only Teamcenter directories affected by Oracle installation. If you created other directories containing data used by Oracle, such as an administration script directory, you should also back up these directories.

### Upgrading an Oracle server

### Upgrade the Oracle server

Upgrade your Oracle server by one of the following methods:

- *Upgrade using the Oracle installer*
- *Upgrade by uninstalling/reinstalling Oracle*

## Upgrade using the Oracle installer

1. Launch the Oracle installer to install a certified version of Oracle server.
2. When the Oracle installer prompts you to upgrade existing databases, enter the required information about the databases you want to upgrade.

Installing an Oracle server is described in the Teamcenter installation guides for Windows and Linux.

## Upgrade by uninstalling/reinstalling Oracle

1. Remove existing Oracle databases.
2. Uninstall all existing Oracle server software.
3. Install a certified version of Oracle server.

   Installation of an Oracle server is described in the Teamcenter installation guides for Windows or Linux.
4. After Oracle installation is complete, import your Teamcenter database from the Oracle dump file into the new Oracle database. Enter the following command on a single line:

   *ORACLE_HOME*`\bin\imp` *db-user/password* `fromuser=`*db-user* `touser=`*db-user* `file=`*file-name*`.dmp log=`*import*`.log`

   Replace *db-user* with the Teamcenter database user account name, *password* with the database user account password, *file-name* with the full path and name of the dump file that contains the exported data, and *import* with the name of the log file.

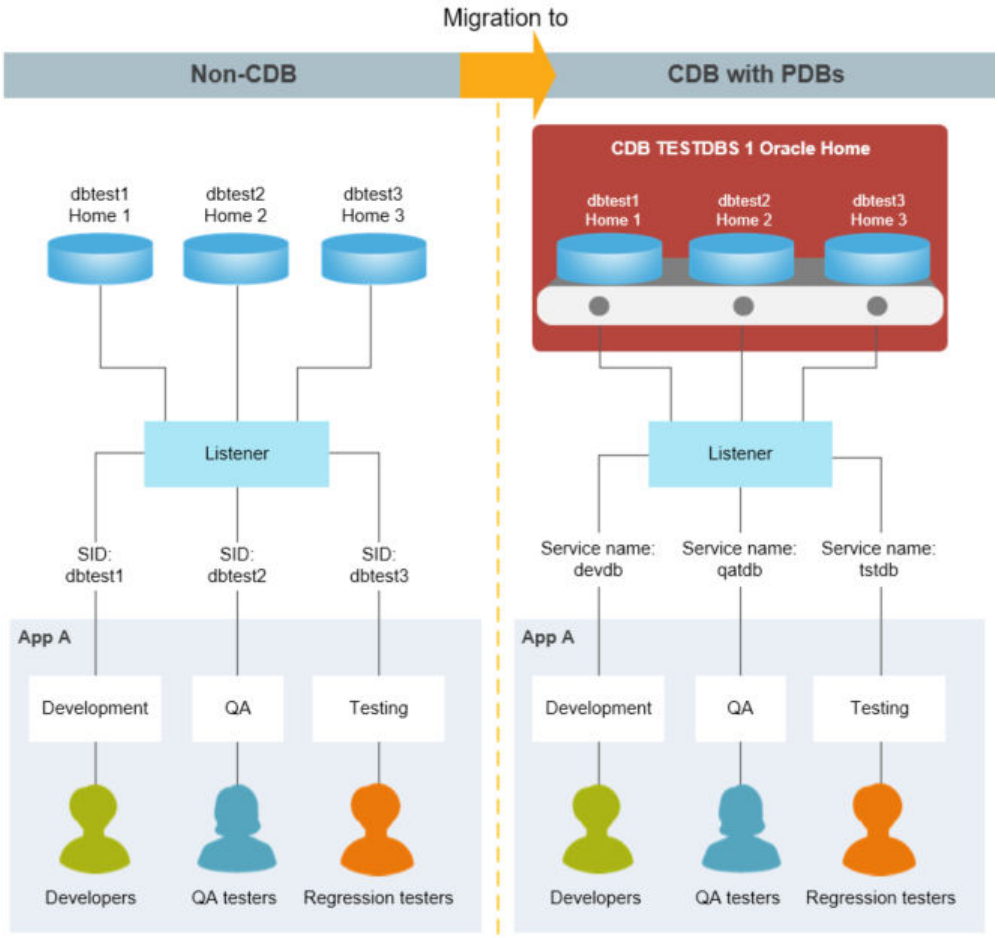
## Migrate a non-CDB database to a CDB database

Teamcenter supports Oracle's **multitenant database architecture** if you use Oracle 12c or later. A multitenant architecture is deployed as a Container Database (CDB) with one or more Pluggable Databases (PDB).

A *Container Database* (CDB) is similar to a conventional (non-CDB) Oracle database, with familiar concepts like control files, data files, undo, temp files, redo logs, and so on. It also houses the data dictionary for objects owned by the root container and those that are visible to databases in the container.

A *Pluggable Database* (PDB) contains information specific to the database itself, relying on the container database for its control files, redo logs and so on. The PDB contains data files and temp files for its own objects, plus its own data dictionary that contains information about objects specific to the PDB. From Oracle 12.2 onward a PDB can and should have a local undo tablespace.

You can **migrate a non-CDB database to a CDB database** using Oracle tools. The following example illustrates the database architectures before and after migration.



Teamcenter supports CDB and non-CDB databases. Be aware that **Oracle has deprecated support for non-CDB databases** and may discontinue support after Oracle 19c.

If you migrate a non-CDB Teamcenter database to a CDB database, you must perform the migration *after* you upgrade to Teamcenter 2412.

## Install Oracle server

You can download and install Oracle from Siemens Support Center if you have purchased it from Siemens Digital Industries Software, or by purchasing it directly from Oracle Corporation.

If you install Oracle from a hard disk, copy the *entire* contents of the Oracle software kit to the hard disk.

You can install Oracle application files on shared directories (on Windows systems) or NFS file systems (on Linux systems). However, Oracle Corporation does not support Oracle database files on shared directories or NFS file systemsan NFS-mounted file system. To ensure data integrity, create database files on local disk drives.

If you install Oracle from an NFS-mounted directory from a remote NFS server, you must execute the installation program on the local server node.

Caution:

- Do not run Oracle Universal Installer as the **root** user.
- Oracle Universal Installer automatically installs the Oracle-supplied version of Java Runtime Environment (JRE). This version is required to run Oracle Universal Installer and several Oracle assistants. Do not modify the JRE except by using a patch provided by Oracle Support Services.

**Windows systems:**

1. Log on to the server host as a member of the Administrators group. If you are installing on a primary domain controller (PDC) or a backup domain controller (BDC), log on as a member of the Domain Administrator group.

   The operating system user account under which you install the Oracle database server must have system administrator privileges.

   The recommended approach is to create a system user account named **oracle** to use during Oracle installation. When you use the **oracle** account to install Oracle, this account is automatically added to the Windows **ORA_DBA** local group, giving it **SYSDBA** privileges.

2. Record the name of the Oracle database server host. Teamcenter Environment Manager requires this name during corporate server installation.

3. In the Oracle RDBMS installation media, launch the **setup** program.

If you install from a DVD, the system displays the **Autorun** dialog box when you insert the DVD.

4. In the **Configure Security Updates** dialog box, specify whether and how you want to be informed about security updates from Oracle, and then click **Next**.

5. In the **Select Installation Option** dialog box, select **Install database software only**, and then click **Next**.

6. In the **Select Database Installation Option** dialog box, select **Single instance database installation**, and then click **Next**.

7. In the **Select Database Edition** dialog box, select the database edition to install, and then click **Next**.

   Teamcenter supports **Enterprise Edition** and **Standard Edition**.

8. In the **Specify Oracle Home User** dialog box, specify the system account you created to install Oracle.

9. In the **Specify Installation Location** dialog box, specify:

   - **Oracle Base**

     Specifies the path in which to install all Oracle software and configuration files.

   - **Software Location**

     Specifies the path in which to install Oracle software files. This is the Oracle home directory.

   Do not install a later version of Oracle into an existing Oracle home directory that contains an earlier version.

10. In the **Perform Prerequisite Checks** dialog box, verify that all the prerequisite checks succeeded and click **Next**.

    If a check fails, review the displayed cause of the failure for that check, correct the problem, and rerun the check.

    A check occasionally fails erroneously, for example, when you install a later patch that obsoletes a listed patch. When you are satisfied that the system meets a requirement, manually verify the requirement by selecting the check box for the failed check.

11. In the **Summary** dialog box, review the information to ensure you have sufficient disk space, and then click **Install**.

12. In the **Install Product** dialog box, monitor the success of the installation stages.

13. When the **Finish** dialog box displays the **The installation of Oracle Database was successful** message, click **Close** to complete the installation.

**Linux systems:**

1. Log on to the server host as the **oracle** user.

2. Record the name of the Oracle database server host. Teamcenter Environment Manager requires this name during corporate server installation.

3. If Oracle was previously installed on the host, search for the following Oracle Net configuration files in the **etc** and **var/opt/oracle** directories and either remove them or relocate them to the corresponding **network/admin** directory in the Oracle home directory:

   **listener.ora**
   **tnsnames.ora**
   **sqlnet.ora**

   This step is required for compliance with the standard of storing Oracle Net configuration files in the **network/admin** directory.

4. Locate the Oracle software kit.

5. If the **/tmp** directory does not have at least 400 MB of free space, set the **TEMP** and **TMPDIR** environment variables to a directory that meets this requirement:

   ```
   $ export TEMP=directory-path
   $ export TMPDIR=directory-path
   ```

   Replace *directory-path* with the path to the directory with sufficient space, for example, **disk/tmp**

6. Start Oracle Universal Installer from the Oracle software kit directory as the **oracle** user:

   ```
   $ umask 022
   $ unset TNS_ADMIN
   $ unset ORACLE_HOME
   $ export ORACLE_BASE=/disk1/oracle
   $ cd $HOME
   $ /mount-directory/runInstaller
   ```

   Replace *mount-directory* with the Oracle software kit directory. This example sets the **ORACLE_BASE** variable to the top level of the Oracle installation.

7. In the Welcome window, click **Next**.

8. If Oracle Universal Installer displays the **Specify Inventory Directory and Credentials** window, enter the directory where you want to install inventory files and the operating system group name for the group that owns the inventory directory; click **Next**.

   Note:

   Siemens Digital Industries Software recommends:

   - Use the default directory (**oraInventory**) in the Oracle base directory.
   - Use the default of the group the **oracle** account belongs to (**dba**).

9. If Oracle Universal Installer prompts you to run the **orainstRoot.sh** script, run it in a separate terminal window as the **root** user and then click **Continue**:

   ```
   $ORACLE_BASE/oraInventory/orainstRoot.sh
   ```

10. In the **Configure Security Updates** dialog box, specify whether and how you want to be informed about security updates from Oracle, and then click **Next**.

11. In the **Select Installation Option** dialog box, select **Install database software only**, and then click **Next**.

12. In the **Select Database Installation Option** dialog box, select **Single instance database installation**, and then click **Next**.

13. In the **Select Database Edition** dialog box, select **Enterprise Edition** and click **Next**.

14. In the **Specify Installation Location** dialog box, specify:

    - **Oracle Base**

      Specifies the path in which to install all Oracle software and configuration files.

    - **Software Location**

      Specifies the path in which to install Oracle software files. This is the Oracle home directory.

    Do not install a later version of Oracle into an Oracle home directory that contains earlier Oracle software.

15. In the **Privileged Operating System Groups** dialog box, specify user groups for the database administrator, operator, and other roles.

16. In the **Perform Prerequisite Checks** dialog box, verify that all the prerequisite checks succeeded and click **Next**.

If a check fails, review the displayed cause of the failure for that check, correct the problem, and rerun the check.

A check occasionally fails erroneously, for example, when you install a later patch that obsoletes a listed patch. When you are satisfied that the system meets a requirement, manually verify the requirement by selecting the check box for the failed check.

17. In the **Summary** dialog box, review the information to ensure you have sufficient disk space and click **Install**.

    If you encounter errors, see the Oracle documentation for troubleshooting information.

18. In the **Install Product** dialog box, monitor the success of the installation stages.

    When the installer displays the **Execute Configuration scripts** dialog box, follow the instructions in the dialog box to run the **root.sh** script in the Oracle home directory. Running this script requires logging on as **root**.

    The **root.sh** script sets the necessary file permissions for Oracle products and performs other **root**-related configuration activities.

19. After the **root.sh** script completes successfully, click **OK** in the **Execute Configuration scripts** dialog box.

20. In the **Finish** dialog box, click **Close** to close Oracle Universal Installer.

### Link the Oracle server to the ODBC library (Linux systems)

Make sure a link exists to the Open Database Connectivity (ODBC) library.

1. Change to the *TC_ROOT*/**lib** directory.

2. Locate the **libodbc** library:

   ```
   ls -la | grep libodbc
   ```

3. Ensure that a link exists between **libodbc.so.2** and **libodbc.so**:

   ```
   ln -s libodbc.so.2 libodbc.so
   ```

4. If the link does not exist, create the symbolic link:

   ```
   ln -s libodbc.so.2 libodbc.so
   ```

## Configure Oracle software

### Configure Oracle Net

Teamcenter uses Oracle Net protocols to communicate with an Oracle database. These protocols require that you run a listener process ( on Linux systems or **OracleTNSListener** on Windows systems) on the Oracle server to listen for remote connect requests and that all clients can translate the service alias identifying the server and database.

On Linux systems, if your site uses Oracle Net Assistant for other databases, Siemens Digital Industries Software recommends that you copy the **listener.ora** and **tnsnames.ora** files containing entries for your designated Teamcenter database and install these copies on the Oracle server. Reload or restart the listener process so that it listens for connect requests to the new database.

On Linux systems, Teamcenter Environment Manager copies the **tnsnames.ora** file and stores it in the Teamcenter data directory. Teamcenter uses the Oracle **TNS_ADMIN** environment variable to locate the **tnsnames.ora** file. However, if the system uses the **TNS_ADMIN** variable to locate configuration files created by Oracle Net Assistant, this setting overrides Teamcenter settings. In this case, you must use Oracle Net Assistant to add entries for Teamcenter databases to existing Oracle Net configuration files.

### Configure Oracle listener

1. Start Oracle Net Manager:

   **Linux systems:**

   In the window in which you started Oracle Universal Installer, start Oracle Net Manager:

   ```
   export ORACLE_HOME=/disk1/oracle/OraHome_1
   $ORACLE_HOME/bin/netmgr
   ```

   **Windows systems:**

   Start Oracle Net Manager. For example, choose **Start→All Programs→Oracle -** *instance-name*→**Net Manager**, or search for **Net Manager**.

2. Create the **listener.ora** file:

   a. Expand the **Local** icon.

   b. Select the **Listeners** folder and choose **Edit→Create**.

   c. Accept the default listener name (**LISTENER**) and click **OK**.

   d. Click the **Add Address** button.

e. Specify the port number.

For the first listener, it is recommended you accept the default port number (1521).

> Tip:
>
> Record the number of the port used by the Oracle database server listener for entry during corporate server installation. Teamcenter Environment Manager requires this port number.

f. In the **Local** tree, click **Profile**.

g. In the **Naming** list (to the right of the **Oracle Net Configuration** tree), choose **General**.

h. Click the **Advanced** tab.

i. In the **TNS Time Out Value** box, type **10**.

This step sets the Oracle server-side **SQLNET.EXPIRE_TIME** parameter. This value determines how often the Oracle server checks for aborted client connections. Teamcenter requires that this parameter be set to a nonzero value, and the recommended value is **10** (10 minutes).

j. Select the **Service Naming** folder and choose **Edit→Create**.

k. Type the **Net Service Name** for your pluggable database and then click **Next**.

l. Select **TCP/IP (Internal Protocol)** and then click **Next**.

m. Enter the host name for your Oracle server and then click **Next**.

If you chose to not use the default port during database creation, change the **Port Number**.

3. Type the **Service Name** and then click **Next**.

4. Click **Test...**

5. Change the **Login** value to the system user name and the **Password** value to the system password used during database installation and then click **Test**.

6. After the connection test is successful, click **Close**.

7. Click **Finish**.

8. Save the listener information, choose **File→Save Network Configuration**.

Oracle Net Manager saves the listener information and creates the **network/admin/listener.oranetwork\admin\listener.ora** and **network\admin\sqlnet.oranetwork/admin/sqlnet.ora** files in the Oracle home directory.

9. Exit Oracle Net Manager, choose **File→Exit**.

10. Start the listener service:

    **Linux systems:**

    In the same window in which you started Oracle Net Manager, start the listener service:

    ```
    $ORACLE_HOME/bin/lsnrctl start LISTENER
    ```

    **Windows systems:**

    In a command prompt, create and start the listener service:

    ```
    cd ORACLE_HOME\bin
    lsnrctl start LISTENER
    ```

    Replace *ORACLE_HOME* with the path to the directory where you installed the Oracle server, for example, **d:\app\tcdba\product\12.2.0\dbhome_1**. This command creates and starts the service if it does not exist. If the service exists, the command starts it.

### Configure Oracle for TCPS

Deployment Center allows you to install the corporate server and the server manager using a TCPS-enabled Oracle database. However, you need to configure the Oracle database for TCPS prior to deployment.

Teamcenter supports the TCPS configuration that uses the Diffie-Hellman anonymous authentication. With Diffie-Hellman anonymous authentication, neither the server nor the client is authenticated through SSL. Authentication must be completed using another method, for example, a user name and password.

In the Oracle database configuration, do not enter a specific cipher suite as this is not supported by Deployment Center.

> Note:
>
> To configure TCPS in Deployment Center, your Oracle server and your Teamcenter corporate server must be installed on Linux machines.

1. In the **listener.ora** section of the Oracle listener machine, set the **SSL_CLIENT_AUTHENTICATION** parameter to **FALSE** as the supported TCPS configuration only covers encryption and data integrity.

2. Verify that the Oracle wallet is stored on the same Teamcenter machine on which the corporate server and server manager are to be installed or upgraded.

When you configure the **Database Server** component In Deployment Center, select **Enable TCPS** and enter the **Wallet Location** (location of the Oracle wallet on the Teamcenter machine) and **TLS Version** (SSL_VERSION specified in the Oracle database).

If you use Quick Deployment, Deployment Center exports these TCPS parameters to the following properties in the configuration file:

| Deployment Center | Quick Deployment Properties |
|---|---|
| **Enable TCPS** | **fnd0_oracleEnableTCPS** |
| **Wallet Location** | **fnd0_oracleWalletLocationTCPS** |
| **TLS Version** | **fnd0_oracleSSLVersion** |

## Create an Oracle database

Create an Oracle database instance with Oracle Database Configuration Assistant (DBCA). Siemens Digital Industries Software provides two templates for creating the Teamcenter database:

- **Teamcenter_Oracle** template is used to create a traditional non-CDB database instance with Oracle user accounts and tablespaces.

- **Teamcenter_Oracle_multitenant** template is used to create a Container and Pluggable database instance where the two databases are identified by their Oracle service names. Teamcenter supports the Oracle 12c multitenant architecture.

The following are key considerations when creating an Oracle Container (CDB) database instance in the Oracle multitenant architecture with Oracle 12c:

- Teamcenter Oracle database tablespaces and the Teamcenter Oracle user account are always created in the pluggable database.

- Teamcenter cannot be installed into the container database.  Attempting to install to a Container database will result in errors during deployment.

- The Teamcenter tablespaces are *not* created using the DBCA template, as this is not supported by Oracle. After you configure the pluggable database, you can manually create a tablespace for the pluggable database, or allow Teamcenter to create the tablespace automatically.

  Using the existing non-CDB template *does* create tablespaces.

For best performance and reliability, database parameters set by Teamcenter templates should be customized to suit your installation. This can be performed by your Oracle administrator after Teamcenter installation is complete.

The deploy script verifies your Oracle version during installation. If your Oracle server does not meet the minimum required version, the installation does not proceed. For information about supported database servers, see the Hardware and Software Certifications knowledge base article on Support Center.

### Copy database creation scripts (Linux systems)

1. Make sure you have access to the Teamcenter software kit.

2. Log on to the Oracle server host as the **oracle** user.

3. Copy the Siemens Digital Industries Software-supplied Oracle database template files:

   a. Access the Teamcenter 2412 software kit.

   b. Copy files from **tc/dbscripts/oracle** to the **templates** directory of the Oracle server:

      ```
      cp /cdrom/tc/db_scripts/oracle/* ORACLE_HOME/assistants/dbca/templates
      ```

4. Open a shell window and set the **ORACLE_BASE** environment variable. For example:

   ```
   export ORACLE_BASE=/disk1/oracle
   ```

   By default, Oracle creates database files in the **oradata** directory in the directory pointed to by the **ORACLE_BASE** environment variable. Before running Oracle Database Configuration Assistant (DBCA), you can set the **ORACLE_BASE** environment variable to the directory where you want database files to reside.

### Copy database creation scripts (Windows systems)

1. Make sure you have access to the Teamcenter software kit.

2. Log on to the Oracle server host as a user who is a member of the **ORA_***instance-name***_DBA** group. This may be the user who installed Oracle on the server host or one assigned to **ORA_***instance-name***_DBA** by a member of the **ORA_***instance-name***_DBA** group.

3. Log on to the Oracle server host as a user who is a member of the **ORA_***instance-name***_DBA** group. This may be the user who installed Oracle on the server host or one assigned to **ORA_***instance-name***_DBA** by a member of the **ORA_***instance-name***_DBA** group.

4. Copy the Siemens Digital Industries Software-supplied Oracle database template files:

   a. Access the Teamcenter 2412 software kit.

   b. Copy files from **tc\dbscripts\oracle** to the **templates** directory of the Oracle server:

      ```
      copy e:\tc\db_scripts\oracle\*   ORACLE_HOME\assistants\dbca\templates
      ```

c. Repeat step **b**, copying files from the same directory on the Teamcenter 2412 software kit.

### Create the Oracle database

1. Make sure you are logged on as the Oracle user.

2. Start Oracle Database Configuration Assistant (DBCA):

   **Linux systems:**

   *ORACLE_HOME*`/bin/dbca`

   **Windows systems:**

   **Start→All Programs→Oracle -** *instance-name*→**Database Configuration Assistant**

   Alternatively, search for **Database Configuration Assistant**.

3. In the **Select Database Operation** dialog box, select **Create a database** and click **Next**.

4. In the **Select Database Creation Mode** dialog box, select **Advanced configuration** and click **Next**.

5. In the **Select Database Deployment Type** dialog box, in the list of templates, select the appropriate template:

   - If you use a non-container (non-CDB) database, select the **Teamcenter_Oracle** template.
   - If you use a container (CDB) database, select the **Teamcenter_Oracle_multitenant** template.

   If you use a CDB database, the DBCA templates do *not* create tablespaces. The template no longer configures tablespaces for pluggable databases.

6. In the **Specify Database Identification Details** dialog box, enter the appropriate values according to the type of database you use:

   - **Container database:**

     a. Accept the default database name in the **Global Database Name** box or type a different name and click **Next**.

        The **SID** box is automatically filled in with the name you enter in the **Global Database Name** box.

> Tip:
>
> Record the SID of the Oracle instance for entry during corporate server installation. Teamcenter Environment Manager requires this name.

b. Select the **Create as Container Database** check box.

The **Create a Container Database with one or more PDBs** radio button is selected by default. Do not change this setting.

c. In the **PDB Name** text box, type the name of the pluggable database, and then click **Next**.

- **Traditional (non-container) database:**

a. Accept the default database name in the **Global Database Name** box or type a different name and click **Next**.

The **SID** box is automatically filled in with the name you enter in the **Global Database Name** box.

> Tip:
>
> Record the SID of the Oracle instance for entry during corporate server installation. Teamcenter Environment Manager requires this name.

b. In the **Database Identification** dialog box, either accept the default database name in the **Global Database Name** box or type a different name and click **Next**.

The **SID** box is automatically filled in with the name you enter in the **Global Database Name** box.

> Tip:
>
> Record the SID of the Oracle instance for entry during corporate server installation. Teamcenter Environment Manager requires this name.

7. In the **Select Database Storage Option** dialog box, select **Use template file for database storage attributes**.

8. In the **Select Fast Recovery Option** dialog box, select the **Specify Fast Recovery Area** check box and accept the default values.

9. In the **Specify Network Configuration Details** dialog box, verify the **listener you created and started** is running and selected in the **Listener Selection** tab.

If the listener is not running, **start the listener** and make sure it is selected before you continue.

10. In the **Select Database Options** dialog box, click **Next**.

11. In the **Specify Configuration Options** dialog box, select **Use Automatic Shared Memory Management**, and then click **Next**.

12. In the **Specify Management Options** dialog box, accept the default selections, and then click **Next**.

13. In the **Specify Database User Credentials** dialog box, select **Use the Same Password for All Accounts**, and then enter and confirm the password.

    The password you enter is applied to the **SYS**, **SYSTEM**, and **PDBADMIN** accounts.

14. In the **Select Database Creation Option** dialog box:

    a. Select **Create Database** check box.

    b. Click **Next**.

15. In the **Summary** dialog box, verify your selections, and then click **Finish** to begin creating the database.

    When the database is created, DBCA displays a window containing information about the created database.

16. In the **Progress Page** dialog box, click **Close** to exit DBCA.

17. After the database is created, check for possible errors in the installation log files:

    **Linux systems:**

    The log files are in the **admin/***SID***/create** directory in the Oracle base directory or, if you did not define the **ORACLE_BASE** environment variable, in the Oracle home directory.

    **Windows systems:**

    The Oracle DBCA displays the directory location of the installation log files in the window that contains information about the created database after the database is created.

If this script did not execute successfully, execute it again using the Oracle SQL*Plus utility. Log on to SQL*Plus as **sysdba**.

The first time Oracle Universal Installer runs, it creates the *ORACLE_BASE***/oraInventory/logs** directory, containing an inventory of installed components and performed actions. The most recent log file is named **installActions.log**. Names of previous installation sessions are in the form **installActions***date-time***.log**. For example:

```
installActions2008-07-14_09-00-56-am.log
```

You can also view a list of installed components by choosing **Installed Products** on any Oracle Universal Installer window. Do not delete or manually alter the Inventory directory or its contents. Doing so can prevent Oracle Universal Installer from locating products you installed on the system.

### Configure the pluggable database

If you use a container (CDB) database, create the Teamcenter Oracle user and set permissions for the pluggable database:

1. Open SQL*Plus and type the following command to connect to the container database:

   ```
   connect user/password;
   ```

   Replace *user* and *password* with the Oracle administrator user name and password. For example:

   ```
   connect system/manager;
   ```

2. Type the following command to set the pluggable database so the Teamcenter Oracle user is created inside the pluggable database.

   ```
   alter session set container=Tc-Oracle-user;
   ```

   For example:

   ```
   connect alter session set container=tcpdb;
   ```

   If successful, SQL*Plus responds:

   ```
   Session altered.
   ```

3. Set privileges for the Teamcenter Oracle user:

   ```
   grant connect, create table, create tablespace, create procedure,
   create view, create sequence, select_catalog_role, alter user,
   alter session to Tc-Oracle-user identified by Tc-Oracle-user;
   ```

   If successful, SQL*Plus responds:

   ```
   Grant succeeded.
   ```

### Create a tablespace for the pluggable database

You can manually create a tablespace for the pluggable database using the following steps. If you do not perform these steps, Teamcenter automatically creates a tablespace with the default size.

1. Open a command prompt and log on to sqlplus as the Oracle administrator, for example, **system**.

2. Create a new tablespace for the pluggable database:

   **create tablespace***tablespace-name***datafile '***dbf-path***/***dbf-filename***' size***dbf-size***M;**

   Replace *tablespace-name* with the tablespace name. Replace *dbf-path*, *dbf-file*, and *dbf-size* with the path, file name, and size of the database file in megabytes. For example:

   **create tablespace tcpdb datafile 'D:\apps\oracle\oradata\tc\tcpdb.dbf' size 100M;**

3. Grant all permissions on the new tablespace to the Teamcenter Oracle **user**:

   a. Enter:

      **alter user***Tc-Oracle-user***quota***dbf-size***M on***tablespace-name***;**

      For example:

      **alter user tcdba quota 100M on tcpdb;**

   b. Enter:

      **grant unlimited tablespace to** *Tc-Oracle-user***;**

4. Log off **sqlplus** by typing **exit**.

## Install and configure Microsoft SQL Server

### Install Microsoft SQL Server

The steps to install Microsoft SQL Server and to configure a database for Teamcenter depend on your operating system, your edition of SQL Server, and your selections during installation.

To optimize MS SQL Server database performance, consider the following steps:

- To implement a Teamcenter network incrementally at multiple sites, configure each site in a Multi-Site Collaboration environment with separate hosts for the MS SQL database server (including Multi-Site Collaboration), the rich client, and volume servers, starting with the first phase. This allows you to configure and manage the network consistently, as you scale it in each phase. You can add CPUs, memory, and disks to the appropriate servers or deploy additional servers as required, without moving or reconfiguring server processes on different hosts or changing operational procedures.

- For large or critical system implementations, implement high-availability systems with mirrored, dual-ported disk arrays. For the Teamcenter volume, consider a file server with storage attached network (SAN) or network attached storage (NAS) disk arrays.

- To minimize system maintenance interruptions, create separate file backup server hosts to process metadata and volume data backups in real time. While the primary disk sets remain online, you can take secondary MS SQL Server and volume disk sets offline simultaneously and back them up together (assuring MS SQL Server and Teamcenter volume synchronization). When the backup is complete, you can return the secondary disk sets online and resynchronize them with the primary disk sets. The file backup servers also serve as fail-over machines.

- To ensure correct character mapping, make sure the database and the Teamcenter server use the same encoding.

For certified versions of MS SQL Server, see the Hardware and Software Certifications knowledge base article on Support Center. **Install the MS SQL Server database server** before you begin installing Teamcenter.

Teamcenter requires the TCP/IP protocol to be enabled, but this protocol is disabled by default when you install Microsoft SQL Server. Before you install Teamcenter, make sure you enable the TCP/IP protocol.

For information about enabling the TCP/IP protocol in Microsoft SQL Server, see **http://technet.microsoft.com**.

### Typical Microsoft SQL Server installation on Windows

1. Log on to an account with system administrator privileges.

2. Launch the Microsoft SQL Server Installation Center application.

3. In the **SQL Server Installation Center** dialog box, click **Installation** in the navigation pane on the left side.

4. Click **New SQL Server stand-alone installation or add features to an existing installation**.

   The SQL Server Installation Center launches the SQL Server Setup wizard.

5. Proceed through the pre-installation tests and other initial setup panes to the **Install Setup Files** pane. Click **Install** to install SQL Server setup support files.

   After setup support files are installed, the wizard displays the **Install Rules** pane. Click **Next**.

6. In the **Feature Selection** pane, select **Instance Features→Database Engine Services** and any other features you want to include.

   Click **Next**.

7. In the **Instance Configuration** pane, select an instance type. Teamcenter supports both **Default Instance** and **Named Instance**.[2]

A default instance in a Microsoft SQL Server installation uses the name **MSSQLSERVER**. Teamcenter's persistent object manager (POM) utilities cannot connect to an instance with this name. If you use a default instance, make sure you connect to the instance using a port connection rather than the name.

If you use a named instance, make sure the instance has a unique name other than **MSSQLSERVER**.

8. Enter remaining instance configuration values, and then click **Next**,

9. Proceed to the **Server Configuration** pane.

   a. Click the **Service Accounts** tab.

   b. Enter account information for starting SQL Server services.

      The SQL Server Setup wizard validates user accounts for SQL Server services. Make sure the accounts you enter exist on the host.

   c. Click the **Collation** tab.

   d. On the **Collation** tab, click **Customize**.

      The wizard displays a customization dialog box for database engine collation.

   e. Select **Windows Collation designator and sort order**.

   f. In the **Collation designator** box, select **Latin1_General** and then select **Binary**.

   g. Click **OK**.

   h. In the **Server Configuration** pane, click **Next**.

10. Proceed to the **Database Engine Configuration** pane.

    a. Click the **Server Configuration** tab.

    b. Under **Authentication Mode**, select **Mixed Mode** and define a password for the SQL Server **sa** logon account.

    c. Specify at least one SQL Server administrator account.

    d. Click **Next**.

---

2 If you choose **Named Instance**, make sure you start the **SQL Browser** service before connecting to the database. If this service is not enabled, you can change these settings using the SQL Server Configuration Manager after installation is complete.

11. Proceed to the **Ready to Install** pane and click **Install** to install.

### Typical Microsoft SQL Server installation on Linux

1. Install a supported version of Microsoft SQL Server on your Linux host.

   For information about installing Microsoft SQL Server on Linux, see Microsoft documentation:

   **https://docs.microsoft.com/**

2. Configure Microsoft SQL Server using Microsoft SQL Server Management Studio.

   Note that the **bulkAdmin** server role is not supported in SQL Server on Linux.

3. After you install Microsoft SQL Server, install the Microsoft Open Database Connectivity (ODBC) driver for Linux as described in **Microsoft documentation**.

4. During installation of the Microsoft ODBC driver, the driver installer prompts you to install the dependent UNIX ODBC driver manager (**unixODBC**). This is a third-party library that you can download from Microsoft. Make sure you install this driver manager.

5. Verify the ODBC driver. After you install the Microsoft ODBC driver and the **unixODBC** driver manager, verify the installation by executing the following command:

   **odbcinst -j**

   This command provides helpful information about the ODBC driver manager configuration. For example:

   ```
   myhost:~> odbcinst -j
   unixODBC 2.3.7
   DRIVERS............: /etc/unixODBC/odbcinst.ini
   SYSTEM DATA SOURCES: /etc/unixODBC/odbc.ini -ILE DATA
   SOURCES..: /etc/unixODBC/ODBCDataSources JSER DATA SOURCES..: /users/
   nvhlwa/.odbc.ini
   SQLULEN Size.......: 8
   SOLLEN Size .......: 8
   SQLSETPOSIROW Size.: 8
   ```

   In this configuration, the ODBC driver manager version is **2.3.7**.

6. Configure the ODBC driver. Open the **odbcinst.ini** file from the location shown in the output of the **odbcinst -j** command. (In the preceding example, this is in the **/etc/unixodbc** directory.) Verify that this file contains a section with the heading `[SQL Server]`.

   If the section does not exist in the file, create the `[SQL Server]` heading and copy the contents of the `[ODBC Driver 17 for SQL Server]` section into it. For example:

```
[ODBC Driver 17 for SQL Server]
Description=Microsoft ODBC Driver 17 for SQL Server
Driver=/opt/mic rosoft/msodbcsql17/lib64/libmsodbcsql-17.5.so.1.1
UsageCount=l
[SQL Server]
Description=Microsoft ODBC Driver 17 for SQL Server
Driver=/opt/microsoft/msodbcsql17/lib64/libmsodbcsql-17.5.so.1.1
UsageCount=l
```

This provides the necessary pointer to the correct driver path.

7. After you complete the installation of Microsoft SQL Server, the Microsoft ODBC driver, and the UNIX ODBC driver manager, you can create a Teamcenter database in Microsoft SQL Server. Create the database during Teamcenter installation through Deployment Center or **using the Teamcenter database template** in Microsoft SQL Server Management Studio.

## Create an SQL Server database

Deployment Center deploy scripts can create and populate a SQL Server database when you install a Teamcenter corporate server.[3] If you want the deploy script to create your Teamcenter database automatically, skip this topic. Otherwise, create your Teamcenter database using the SQL Server Management Studio.

1. Make sure you have access to the Teamcenter software kit.

2. Launch Microsoft SQL Server Management Studio.For example, on Windows systems:

   **Start→Programs→Microsoft SQL Server→SQL Server Management Studio**

   Alternatively, search the start menu for **SQL Server Management Studio**.

3. In the SQL Server **Connect to Server** dialog box, log on using the system administrator (**sa**) logon name and password.

4. Choose **File→Open→File** or press Control+O.

5. Browse to the **tc\db_scripts\mssql** directory (on Windows systems) or the **tc/db_scripts/mssql** directory (on Linux systems) in the Teamcenter software kit.

6. Select the **create_database.sql.template** file and click **Open**.

   If SQL Server Management Studio prompts you to log on, enter the system administrator (**sa**) logon name and password.

3 In the **Database Server** component, you can enter information for the SQL Server database. To create a new database, enter new values. To connect to an existing database, enter values for the existing database. For information about installing a corporate server, see *Create a Teamcenter environment using Deployment Center*.

7. Edit the database template (**create_database.sql.template**) to replace the necessary values.

   The following table describes the database parameters to replace in the template. Within the template file, there are also comments on values that must be replaced.

| Parameter | Example value | Description |
|---|---|---|
| **@DB_NAME@** | **TC** | Name of the database to create. |
| **@DATA_PATH@** | **D:\MSSQL_DATA** (Windows systems) or **/mssql_data** (Linux systems) | Path to the directory in which to place the data file. |
| **@USER_NAME@** | **tcdba** | Database logon name for the Teamcenter database. |
| **@PASSWORD@** | **tcdbapw** | Password for the database logon name. |
| **@COLLATION@** | **Latin1_General_BIN** | Collation sequence you want the Teamcenter database to use. Choose the **appropriate collation for your locale**. The collation value must end with **_BIN**.[4].<br><br>*Collation* defines the alphabet or language whose rules are applied when data is sorted or compared. The collation value determines the character set used by the database server. |
| **@LANGUAGE@** | **us_english** | Database language. |

8. Save the newly modified file as *filename***.sql**, removing the **_template** extension.

9. Open the new file in Microsoft SQL Server Management Studio.

10. In the SQL Editor toolbar, click **Execute** (or choose **Query→Execute** to begin creating the database.

11. When creation of the MS SQL database instance is complete, verify the newly created database. In the **Object Explorer** pane, under the MS SQL Server host name, expand the **Databases** tree. Verify the new database name is included in the list of databases.

4 Do not use the default collation value that ends with **_CI_AS**.

# 7. Install the Siemens License Server

Before you install Teamcenter, you must download and install the supported version of the **Siemens License Server** to distribute licenses to Teamcenter hosts.

For the version of the Siemens License Server certified with Teamcenter 2412, see the Hardware and Software Certifications knowledge base article on Support Center.

Download and install the Siemens License Server:

1. Open Support Center:

   **https://support.sw.siemens.com**

2. Under **Product Centers**, find **Siemens License Server**.

   > Caution:
   >
   > Make sure you download **Siemens License Server**, *not* **Siemens PLM Licensing**.[1]

3. In the Siemens License Server product center, click **Downloads**, and then download the certified version of the Siemens License Server.

4. Install the License Server according to the *Siemens Digital Industries Software License Server Installation Instructions* available from the Siemens License Server downloads page.

5. On your designated Teamcenter corporate server host, set the following system environment variables:

   **SPLM_LICENSE_SERVER**

   Set to the location of the Siemens License Server:

   *port@host*

   Replace *port* with the port number and *host* with the machine name of the License Server, for example, **29000@tchost**.

   **TCP_NODELAY**

   Set to a value of **1** on the License Server host. This helps optimize logon time when launching Teamcenter.

1 Siemens PLM Licensing is no longer supported by Teamcenter. The Siemens License Server is the currently supported license server.

6. Install Teamcenter licenses on the License Server according to the information provided to you by Siemens Digital Industries Software support.

The **Siemens License Server downloads page** contains additional links to documentation, Knowledge Base articles, and videos about installing and maintaining the License Server.

> Caution:
>
> The License Server must be running and two or more seats must be available on that license server during Teamcenter server installation. Otherwise, database creation fails because the **make_user** utility cannot create the required users in the database.

# Part II: Build the Teamcenter Environment


Create a Teamcenter environment. Install a corporate server, the central component of the Teamcenter environment. Distribute Teamcenter software components across your network to optimize performance and security. Add optional applications that provide the specialized capabilities and integrations to your users.

Install Teamcenter and Active Workspace in a test environment, including the applications and components you want to use.

Installing a test environment allows you to configure components and identify and resolve and potential issues before you **deploy** your settings to a production environment.

# 8. Configure available units of measure

Administrators can configure the units of measure that are available across Teamcenter. This unit management system (UMS) based on classified units is first available in Teamcenter version 2406.

This procedure is not required, but if you want to include optional or custom units of measure in your Teamcenter environment, you can add them before you begin installation.

A list of nearly 1400 units of measure is provided in the Teamcenter installation kit. Out of the box, only a subset of the units are installed, depending on the applications installed. Installers and administrators can configure the units to add, and can add new units.

1. In the Teamcenter kit or in the *TC_ROOT***TD** folder, find and open *unit_definitions.csv*.

   Application columns begin with column Q (**BASE**). Unit rows for which an application column has the value **1** will be considered by the ums_import_unit_definitions utility.

   > Note:
   >
   > During initial installation, if an application is installed in the environment, unit rows for which the application column has the value **1** are imported to the database.

2. Edit the file to ensure that the unit definitions you want to administer have the value **1** for an application.

   In the event that no class definition exists for a unit you want in your environment, create a new row.

3. Using the edited *unit_definitions.csv*, run the ums_import_unit_definitions utility.

# 9. Create user accounts and directories

Create the required user accounts and directories that Teamcenter requires for installation and maintenance.

## Create required user accounts

On the local host where you install Teamcenter software, create the Teamcenter operating system user account.

All Teamcenter services run as this user account. On Windows systems, make sure this account belongs to the Administrators group and is granted the **Log on as a service** right.

Ensure that all Teamcenter directories and files are owned and writable by this operating system user.

- **Operating system logon account**

  Create an operating system logon account for Teamcenter. On Windows systems, this account must belong to the **Administrators** group and must be granted the **Log on as a service** right. Teamcenter services run on the server as this user account.

  Log on using this account when you install the Teamcenter environment and when you perform maintenance. Ensure that all Teamcenter directories and files are owned and writable by this operating system user.

  The following are some services that may run under this user account:

  | | |
  |---|---|
  | Database Daemons | Schedule Manager |
  | FSC | Dispatcher Module and Client |
  | Linked Data Framework Web Services | AM Read Expression Service |
  | Multi-Site IDSM and ODS daemons | Server Manager |

- **Teamcenter administrative user account**

  Teamcenter provides an administrative user account named **infodba**. Teamcenter Environment Manager automatically creates this account when you install Teamcenter on a server host. This account is used by the Teamcenter administrator to access the Teamcenter system administration functions to perform setup and maintenance tasks. You create a password for this account during Teamcenter installation.

Caution:

- The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab.

  In addition, the password must not contain any of the following characters:

  **! # @ $ % = & ' " ^ : ; . _ < > ( ) { }**

- Never use the **infodba** user to create working data or initiate workflow processes. The **infodba** user is to be used *only* for specific installation tasks described in Teamcenter installation documentation. Using this account to create data or initiate workflow processes can cause unexpected and undesirable behaviors.

  If you require a user with high-level privileges to create data, create a new user and grant database administrator privileges to that user.

- **Database user**

  Create a database user to be the owner of Teamcenter-created tables and to perform tasks required by Teamcenter. You create this database user either by using the templates provided for Oracle databases, or by using Teamcenter installation tools to populate a database. Teamcenter installation tools refer to this user as **DB user**.

  On Linux systems, if Oracle and Teamcenter applications or files are shared using NFS/CIFS, you must standardize the user and group IDs of the Teamcenter and Oracle accounts to give them the same access privileges on all systems.

  Each user and group is identified by an alphanumeric name and an ID number. The ID number is retained with the file information when a file is exported across a network. If the ID numbers do not match for a user or group, file access privileges may be unintentionally granted to the wrong user, or not granted at all, on an NFS/CIFS client.

## Create required directories

**Teamcenter installation root directory**

Choose a parent directory to contain Teamcenter software. This parent directory must exist before installation. The Teamcenter root directory is created within this directory during installation. Requirements for this directory:

**Windows systems:**

- The directory must be excluded from real-time virus scanning.

  Real-time virus scanning prevents Teamcenter from updating the persistent object manager (POM) schema during installation, causing installation errors.
- If the directory is on a mapped drive or a UNC path (not on the local host) you must be logged on as an authenticated domain user to ensure the remote host recognizes

you. Alternatively, you can set the permissions on the remote host to allow an anonymous user to access it. This is necessary to ensure Teamcenter services such as the FMS server cache (FSC) and Multi-Site Collaboration services can start.
- The directory must be on an NTFS partition, not a FAT partition. This is necessary to take advantage of the file security features of NTFS.

**Linux systems:**

- If the directory is in an automounted NFS directory, but you must supply the automount link name for the Teamcenter application root directory. Do not supply the automounted directory (for example, **/tmp/mnt/***node-name*).
- If you install File Management System file caches and/or Multi-Site Collaboration services, the directory must be on a local disk.

**Teamcenter volume location**

Choose a parent directory to contain a Teamcenter volume or volumes.

This parent directory must exist before installation. The volume directory is created within this parent directory during installation.

Do *not* place the volume directory under the Teamcenter application root directory. Doing so can cause problems when upgrading to a new version of Teamcenter.

# 10. Create a Teamcenter environment using Deployment Center

Create a Teamcenter environment with default applications and components using Deployment Center.

Create a new Teamcenter environment with common Teamcenter components by performing the following tasks:

1. Add Teamcenter software to the repository.
2. Create an environment and choose software.
3. Choose options.
4. Choose applications.
5. Choose components.
6. Configure components.
7. Deploy the environment.

## Add Teamcenter software to the repository

1. Expand the Teamcenter 2412 software kit. Copy the unzipped contents to the *software* subdirectory in one of your registered repository locations.

2. Log on to Deployment Center, and click **SOFTWARE REPOSITORIES**.

   The **Software Repositories** page opens the **Active Media** tab of the repository and displays the **Software Media** table.

3. Verify that the added software appears in the list of available software. The list may take a few minutes to update.

   If the software does not appear in the **Software Repositories** page, inspect the repository log files for repository scanning issues or software file problems. The repository log files are in the **webserver\repotool\logs** directory on the Deployment Center server.

## Create an environment and choose software

1. In Deployment Center, click **ENVIRONMENTS**.

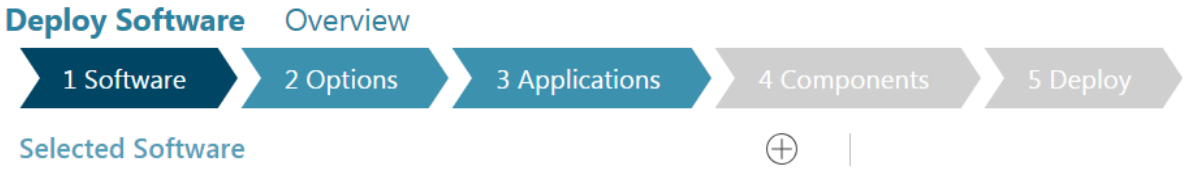
2. On the far right below the command bar, click **Add Environment**⊕.

   The new environment appears highlighted in the **All Environments** list.

3. To view properties of the new environment, choose **Overview**.

   If you want to edit properties such as **Name** and **Type**, click **Start Edit**. To save changes, click **Save Edits**. To cancel changes, click **Cancel Edits**.

4. Choose **Deploy Software** to return to the **Software** tab.

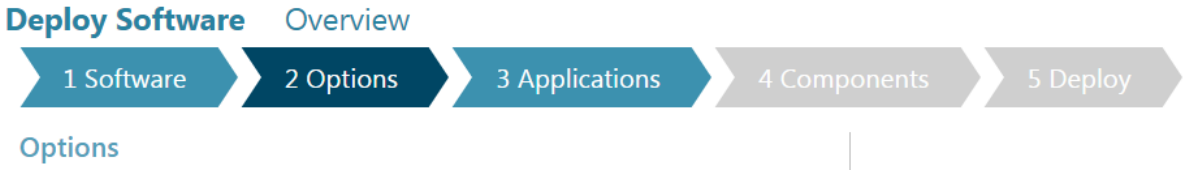



5. In the **Available Software** panel, select the Teamcenter 2412 software, and then click **Update Selected Software**.

   If the software you need is not listed, you must **add it to the software repository**.

6. Proceed to the **Options** tab.

## Choose options




In the **Options** tab, choose deployment options for your environment.

1. Choose the **Environment Type**.

   **Single Box** Installs all components on a single machine.

   This environment type is useful for developing and testing Teamcenter deployment.

   After you define **Machine Name**, **OS**, and **Teamcenter Installation Path** parameters for one component in the environment, those values are inherited by the other components. Changing these parameters for any component changes them for all components.

   **Distributed** Enables installation of components on separate machines.

   This environment type is common for production environments.

   **Machine Name**, **OS**, and **Teamcenter Installation Path** configuration values are shared only with other components that are required to be on the same machine. When configuring a new component, you can select a **Machine Name** from the dropdown list or enter a new machine name.

   You can change the value from **Distributed** to **Single Box** if an installation or an update is not in progress. For configured components that are not yet installed, **Machine Name**, **OS**,

and **Teamcenter Installation Path** are changed to the values specified for the corporate server component.

2. Choose **Architecture Type**.

| | |
|---|---|
| **Java EE** | Filters the available components to those that support the Java EE architecture. |
| **.NET** | Filters the available components to those that support the Windows .NET architecture. |

If you have already deployed your environment with one of these architectures, the architecture type is set and cannot be changed.

3. Choose **Infrastructure Type**.

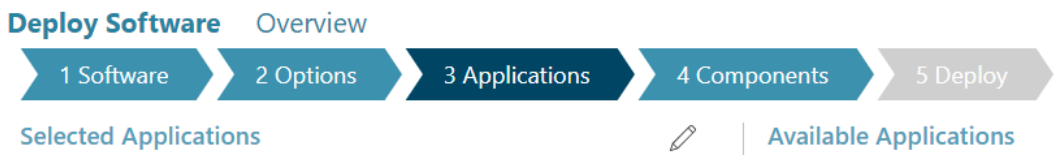
| | |
|---|---|
| **Local** | Specifies an environment in which server and client components connect to the current environment. Also, mass client components shared from a Global infrastructure can be imported into a Local infrastructure. This is the default selection in a new environment. |
| **Global** | Specifies an environment in which components can be shared to multiple environments, and with those environments' databases. A Global infrastructure is used to define mass client information that can be shared to multiple environments managed in Deployment Center.<br><br>Support for global infrastructure is limited to client components such as the four-tier rich client, TCCS, and Security Services Session Agent. |

4. If you install SolrCloud in a distributed environment with high availability, select the **High Availability** check box.

> Note:
>
> This option applies only to SolrCloud. It does not apply to high availability configuration for other components.

5. When your selections are complete, click **Save Environment Options** to proceed to the **Applications** tab.

## Choose applications



In the **Applications** tab, the **Selected Applications** panel displays applications preselected by default.

The list of available applications and the default selections are based on the software you selected in the **Software** tab. Because you selected Teamcenter software, the **Teamcenter** application group is selected, which contains **Teamcenter Foundation** and essential Active Workspace applications.

To continue with default Teamcenter applications only, proceed to *Choose components*. Otherwise, to modify the selected applications, perform the following steps:

1. In the **Applications** tab, click **Add or Remove Selected Applications**.

   The **Available Applications** panel displays the applications available to install.

2. In the **Available Applications** list, edit the selected applications:

   - **Add or remove applications**

     Select an individual application to add it to your environment, or deselect it to remove it from your environment.

   - **Add or remove application groups**

     Select or deselect an application group to add or remove *all* applications in that group.

   - **Clear the selected applications list**

     Select and then deselect the top-level **Teamcenter** application group. This removes all default applications and any selections you made.

   Expand or collapse application groups to simplify navigation. To search for an application by name, use your web browser search.

   Some application names have changed since previous releases. See *Application names changed in Deployment Center* for more information about application names.

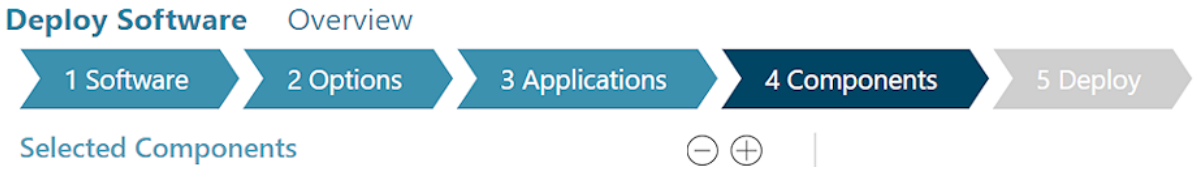
   You can further **add applications** to your environment after you complete creating and deploying your environment.

   > Note:
   >
   > Do *not* select the **Teamcenter Rapid Start Configuration** or **Teamcenter Rapid Start Active Workspace**. These applications are supported only in Teamcenter Rapid Start environments.

3. Click **Update Selected Applications** to save your changes to the **Selected Applications** list.

When you are satisfied with your **Selected Applications** list, proceed to the **Components** tab.

## Choose components




In the **Components** tab, you configure components for installation. The **Selected Components** list displays components that are automatically added by the applications in the **Selected Applications** list.

To continue with default components only, proceed to *Configure components*. Otherwise, to add components, perform the following steps:

1. Click **Add component to your environment** ⊕ to display the list of **Available Components**.

   Available components are determined by your selected software and applications. If a component you want is not listed, modify your selections in those tabs. To search for a component, use your web browser search. Expand or collapse component groups to simplify navigation.

2. In **Available Components**, select components to install, and then click **Update Selected Components** to add them to the **Selected Components** list.

   For information about a component, see its **DESCRIPTION** in the **Available Components** panel.

3. Observe the configuration status of selected components.

   The **COMPLETE** column displays the completion state for each component. At this time in the process, the **Deploy** tab is disabled because selected components have not been fully configured. The **Deploy** tab is enabled when the required parameters for all components are **100%** complete.

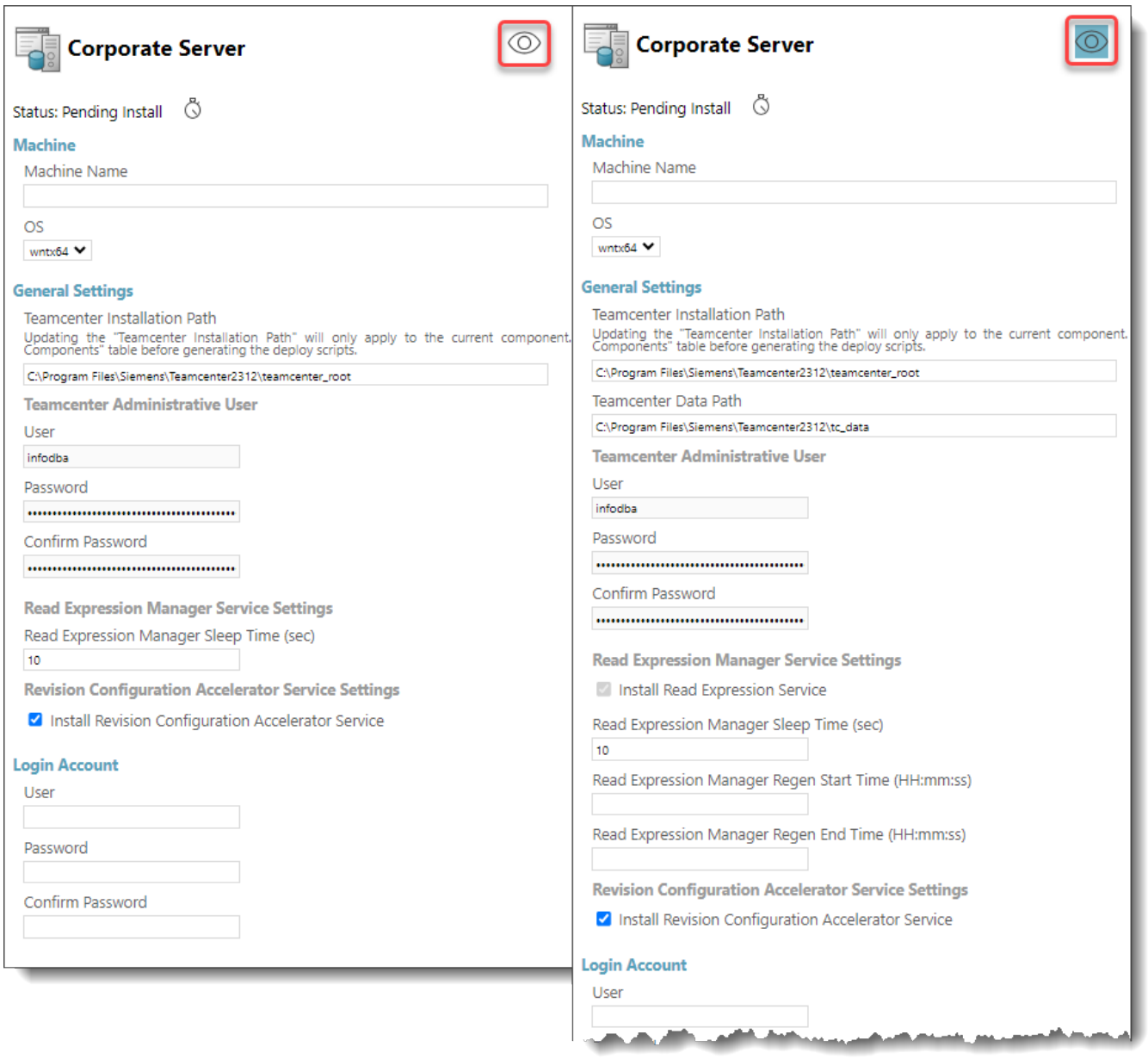
   When you are satisfied with the selected components, proceed to *Configure components*.

## Configure components

Click a component in the **Selected Components** panel to view its configuration parameters. Parameters for a given component can be displayed in two views:

| | |
|---|---|
| | **Show all parameters**<br><br>Required parameters view displays only required parameter information. Click to expand the view to display both required and optional parameters. |
| | **Show only required parameters** |

All parameters view displays both required and optional parameter information. Click to collapse the view to required parameters.




Select each component in your Teamcenter environment in turn, configure required parameters, and then click **Save Component Settings**. Repeat these steps for each component until all components are fully configured, showing a value of **100%** in the **COMPLETE** column.

Note the following behaviors as you set parameters:

- **You can save component settings in progress**

If you do not have values for all required parameters, you can save your component settings at any time and return to finish them. However, the **Deploy** tab will remain disabled until all components are **100%** configured.

- **Some parameters inherit from others**

  As you configure components, you may observe some components display a status of **100%** even though you have not selected them.

- **Optional parameters remain available**

  After a component displays a status of **100%**, you can still select that component to review or change parameter settings, or set additional optional parameters.

- **Machine parameters are synchronized in Single Box environments**

  In **Single Box** environments, the **Machine Name**, **OS**, and **Teamcenter Installation Path** parameters are automatically copied from the first component you configure, and any changes to these parameters are copied to other components. An exception is the **Database Server** component, which assumes a separate machine with a preexisting database server. For other components, Deployment Center ensures that Single Box components specify the same machine.

  In a **Distributed** environment, components can specify distinct machine parameters.

Configure the required parameters for each of the following default components in your Teamcenter environment. Select each component in turn, enter values for the required parameters, and then click **Save Component Settings**. Repeat these steps for each component until all components are fully configured, showing a value of **100%** in the **COMPLETE** column.

The following components are included in a default Teamcenter environment with the **Single Box** environment type and **Local** infrastructure type. If you add applications, more components may be included.

**Active Workspace Client Builder**
**Active Workspace Gateway**
**Container Configuration***(Linux machines only)*
**Corporate Server**
**Database Server**
**FSC**[1]
**FSC Group**
**FSC Keys**
**HTTPS Config**
**Indexer**
**Indexing Engine**
**Licensing server**
**Microservice Node**
**Server Manager**
**Server Manager Cluster Configuration**
**Teamcenter Vault**
**Teamcenter Web Tier (Java EE)***(Java EE architectures only)*
**Teamcenter Web Tier (.Net)***(.NET architectures only)*

1 FMS server cache.

### Active Workspace Client Builder

| Value | Description |
|---|---|
| **Publish Active Workspace Client Assets** | Specifies you want to automatically publish Active Workspace content to the Gateway. |

For more information about installing the Active Workspace client installation, see *Install the Active Workspace client*.

**Return to components list**

### Active Workspace Gateway

| Value | Description |
|---|---|
| **Port** | Enter the port for Active Workspace Gateway. The default value is **3000**.<br><br>The URL to the Active Workspace client interface will use this port. |
| **URL Prefix** | Specifies an optional prefix for Active Workspace Gateway. For example, if you work with load balancers, you may need to change the URL prefix for your site to a non-root context. |
| **Use SSL protocol** | Specifies the protocol you want the Active Workspace Gateway to use:<br>**https** — Active Workspace Gateway uses the HTTPS configuration settings from the **HTTPS Config** component on the Active Workspace Gateway machine.<br>**http** — Active Workspace Gateway uses the HTTP protocol. |
| **Gateway URL** | Specifies the URL to the Active Workspace Gateway. This value is constructed from other parameters and is not directly editable. It has the following form:<br><br>*protocol***://***machine***:***port*<br><br>For example:<br><br>**https://myCorp:3000** |

For a full description of Active Workspace Gateway installation, see *Install Active Workspace Gateway*.

**Return to components list**

### Container Configuration*(Linux machines only)*

| Parameter | Description |
| --- | --- |
| **Container Registry URL** | Enter the machine name or IP address and port of the container registry. Do not enter a protocol. |
| **Container Repository Name** | Enter the name of the repository for Teamcenter microservices.<br>A repository is a logical grouping of container images within the registry. The repository name must exist in the container registry before you run the scripts generated by Deployment Center. The recommended name is **teamcenter**. |
| **Container Manager** | Choose one of two container manager types, **Docker Swarm** or **Kubernetes**.<br>For Kubernetes, specify the **Namespace**. A namespace is the unique name that identifies the group of Teamcenter resources interacting with each other in a Kubernetes cluster. The value you enter replaces placeholders in microservice **.yml** files. This is the same namespace described in the procedure **Deploy microservices in Kubernetes**. |

For a full description of microservice deployment, see *Microservices and the microservice framework*.

**Return to components list**

### Corporate Server

If you create a Single Box environment, set the **Machine Name**, **OS**, and **Teamcenter Installation Path** on a core component such as the **Corporate Server**. The **Teamcenter Installation Path** specifies the Teamcenter root directory (*TC_ROOT*) on each given component machine. This path must meet the **requirements for the Teamcenter root directory**.

| Parameter | Description |
| --- | --- |
| **Teamcenter Administrative User** | During a corporate server installation, the user name and password for the **Teamcenter Administrative User** are read-only. You must change the password for this account after installation. |
| **Read Expression Manager Sleep Time** | Specifies the time in seconds for the Read Expression Manager service to wait until a new update tab is performed. The default is **10** seconds. |
| **Login Account** | Specifies the user name and password for the operating system account under which you install Teamcenter. |

If your environment is distributed, Deployment Center may automatically add additional **business logic servers** as needed to support distributed components.

**Return to components list**

### Database Server

| Parameter | Description |
|---|---|
| **Database Creation Settings** | Options for creating the Teamcenter database. |
| **Create and populate database. Create new data directory.** | Choose this option if no Teamcenter database or data directory exists and you want Deployment Center to create both.<br>Specify the **Database Path**.<br>For Oracle databases, this specifies the location of the tablespaces for the Teamcenter database on the Oracle server. This is typically*ORACLE_HOME***oradata***Oracle_SID* (on Windows systems) or *ORACLE_HOME*/**oradata**/*Oracle_SID* (on Linux systems).<br>For Microsoft SQL Server databases, this specifies the directory in which to create the Teamcenter database on the SQL Server server. |
| **Populate database. Create new data directory.** | Choose this option if a database exists but is not populated with Teamcenter data. You want Deployment Center to populate the database and create a new data directory. |
| **Copy Environment using existing populated database.** | Choose this option if a database exists and is populated. You want Deployment Center to use this database and create a new data directory.<br>In **Volume Information**, click **Add Row**, and then type the **VOLUME NAME** and **ORIGINAL HOST** of the database you want to copy from, and a **COPIED VOLUME PATH** for the new data directory. |
| **Database Settings** | Settings for the Teamcenter database. Enter settings according to your database type. |
| | *Oracle databases* |
| **Database Server** | Select **Oracle**. |
| **Oracle Database User** | Specifies a database user name:<br>• If you chose the first option under **Database Creation Settings**, type the name of the new database user you want to create.<br>• If you chose the second or third options under **Database Creation Settings**, enter the name of the existing database user for the database you want to use. |
| **Password** | Specifies the password for the Oracle database user. Type the password again in **Confirm Password**. |
| **Service** | Specifies the name of the service for the Oracle instance. The default name is **tc**.<br>The service name was determined when the Oracle server was installed. |
| **Port** | Specifies the number of the port on which the Oracle server listens. The default value is **1521**.<br>The port number was determined when the Oracle server was installed. |

| Parameter | Description |
| --- | --- |
| **Enable TCPS** | Specifies whether to your **Oracle server is configured** for secured communication using TCPS protocol. If TCPS is enabled, select the **Enable TCPS** check box and then type values for **Wallet Location** and **TLS Version**. |
| **Wallet Location** | Specifies the location of the wallet on the Teamcenter machine where Oracle wallets are kept.<br>Note:<br>To configure TCPS in Deployment Center, your Oracle server and your Teamcenter corporate server must be installed on Linux machines. |
| **TLS Version** | Specifies the version of Transport Layer Security (TLS) configured on the Oracle server. This is equal to **SSL_VERSION** value specified on Oracle database machine. |
| *Microsoft SQL Server databases* | |
| **Database Server** | Select **MSSQLServer**. |
| **User** | Specifies a database user name:<br>• If you chose the first option under **Database Creation Settings**, type the name of the *new* database user you want to create.<br>• If you chose the second or third options under **Database Creation Settings**, enter the name of the *existing* database user for the database you want to use. |
| **Password** | Specifies the password for the database user.<br>Specifies the password for the database user. Type the password again in **Confirm Password**. |
| **Port** | If you connect to Microsoft SQL Server using a specific port, choose this option and enter the **Database Port** number you specified when you installed MS SQL Server. |
| **Instance** | If you connect to Microsoft SQL Server using a named instance, choose this option and enter the **Named Instance** name you defined when you installed MS SQL Server. |
| **Database Name** | Specifies the name of the MS SQL Server database. The default name is **tc**.<br>The database name was determined when database was created. |
| **Collation** | Specifies the collation used by the Teamcenter database on the Microsoft SQL Server server. *Collation* defines the alphabet or language whose rules are applied when data is sorted or compared. |
| **Enable UTF Mode?** | Specifies whether to enable support for UTF-8 encoding in the Teamcenter database. |

| Parameter | Description |
|---|---|
| | Microsoft SQL Server does not provide native support for UTF-8. The **Enable UTF-8** option enables the Teamcenter server to convert character encoding to and from UTF-8 when interacting with the database.<br><br>To use UTF-8, you must **configure your machine to support UTF-8** before you install Teamcenter host to support UTF-8.<br><br>Specifies the password for the Oracle system administrator account.<br><br>The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab. In addition, the password must not contain any of the following characters:<br><br>**! # @ $ % = & ' " ^ : ; . _ < > ( ) { }** |
| **Database System User Credentials** | Database system credentials. These parameters are enabled if you chose the first option under **Database Creation Settings**: |
| **User** | Specifies the user name of the database system administrator account. For Oracle databases, the default value is **system**, for Microsoft SQL Server databases, the default value is **sa**. |
| **Password** | Specifies the password for the database system administrator account.<br><br>The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab. In addition, the password must not contain any of the following characters:<br><br>**! # @ $ % = & ' " ^ : ; . _ < > ( ) { }** |

Return to components list

**FSC**

| Parameter | Description |
|---|---|
| **Login Account** | Specifies the user account under which the FMS server cache (FSC) service runs. Choose one of the following options:<br><br>• **This Account**<br><br>Specifies you want the FSC service to run under a specific user account. If you choose this option, type the credentials for the account:<br><br>**User** — Specifies user name or the domain and user name for the account, for example, **domain\user**.<br><br>**Password** — Specifies the password for the designated user account.<br><br>• **Local System Account** |

| Parameter | Description |
| --- | --- |
| | Specifies you want the FSC service to run under the current local system user account (the account under which you run the deploy script). |
| **FSC Master Settings** | A Teamcenter network must have at least one primary (master) FSC. If you want to designate the current FSC as an FSC primary, select the **Is Master?** check box. Otherwise, type the URL to the parent FSC in the **FSC Parent URL** box. |

For an introduction to File Management System (FMS) components, see **Overview of FMS installation**.

For detailed information about FMS deployment, see *Teamcenter Administration*.

**Return to components list**

### FSC Group

| Parameter | Description |
| --- | --- |
| **Instance** | Specifies an instance name for the FSC group. |
| **FSC Group Name** | Specified the name of the FSC group. An FSC (FMS server cache) group is a group of server caches defined in the File Management System (FMS) master configuration file. |

For an introduction to File Management System (FMS) components, see **Overview of FMS installation**.

For detailed information about FMS deployment, see *Teamcenter Administration*.

**Return to components list**

### FSC Keys

| Parameter | Description |
| --- | --- |
| **Generate New Keys** | Specifies you want to generate new keys |
| **Key Store Password** | A Teamcenter network must have at least one primary (master) FSC. If you want to designate the current FSC as an FSC primary, select the **Is Master?** check box. Otherwise, type the URL to the parent FSC in the box. |
| **Use Symmetric Keys** | Specifies you want to use symmetric keys instead of asymmetric keys. |
| **Configure Key Alias?** | Specifies you want to use a key alias. Enter the key alias under which you want to store the FMS key in the **Key Alias** box. |
| **Configure Key Alias Password?** | Specifies you want to use a key alias password. Enter and confirm the **Key Alias Password**. |

| Parameter | Description |
|---|---|
| **Sync FSC Key for Multi-Site?** | Specifies you want to use a symmetric key for Multi-Site. Enter and confirm your **Multi-Site symmetric key**. |
| **Use Asymmetric Keys (advanced)** | Specifies you want to use asymmetric keys instead of symmetric keys. |

For detailed information about configuring FMS ticket signing keys, see *Teamcenter Administration*.

Return to components list

### HTTPS Config

| Parameter | Description |
|---|---|
| **Use Deployment Center Vault Generated Certificate** | Specifies you want to use a certificate issued by the Teamcenter vault for HTTPS communication with the given machine. |
| **Use Specified Certificate** | Specifies you want to use a certificate issued by a third-party certificate authority (CA) for HTTPS communication with the given machine.<br>Make sure your certificate is not an intermediate certificate but contains the *entire* certificate chain. |
| **Private Key Path Input** | Specifies the location of your SSL private key on the given machine. |
| **Certificate Path Input** | Specifies the location of your SSL certificate on the given machine. |
| **Root CA Path** | Specifies the path to the root certificate issued by a certificate authority (CA). |

In a distributed environment, for each machine on which you specify HTTPS protocol for an installed component, Deployment Center adds a separate instance of the **HTTPS Config** component. The certificate paths you specify must exist on the given machine.

In a single-box environment, Deployment Center adds only a single instance of the **HTTPS Config** component.

Return to components list

### Indexer

| Parameter | | Description |
|---|---|---|
| **Install Database Triggers for Indexing** | | Select the **Install Database Triggers for Indexing** check box if you want to install database triggers. |
| **Maximum Teamcenter Connections** | | Specifies the maximum number of connection between the Teamcenter server and the indexer to be open at a given time. This value should be less than the number of warmed up Teamcenter servers available in the server manager. The default value is **3**. The minimum value allowed is **1**. |
| **Staging Directory** | | (For Dispatcher-based indexing only)<br>Specifies is Dispatcher staging directory. This directory is defined in the **Dispatcher Components** panel when you install the Dispatcher server.<br>Specifies the staging directory used by the indexer. In standalone indexing mode, this directory is in the location where the standalone indexer is installed. In Dispatcher-based indexing mode, this is usually the same as the Dispatcher server staging directory. |
| **Install Indexer as a Service?** | | Select the **Install Indexer as a Service** check box if you want to install the objdata synchronization flow and the suggestion builder synchronization flow of the indexer as services. |
| | **Service Name** | The **Service Name** fields populate with suggested names for the services, and can be edited. |
| | **Sync interval** | The **Sync Interval** fields populate with suggested intervals for the synchronization flows and can be edited. |
| | **Start Service** | Select the **Start Service** check box to automatically start the service. |
| | **Service Name** | Specifies the display name for the Suggestion Builder Service. |
| | **Sync interval** | Specifies the sync interval for the Suggestion Builder Service. |
| | **Start Service** | Select the **Start Service** check box to automatically start the Suggestion Builder Service. |
| **Operating System User** | | Settings to configure the OS user name under which Indexer services run. |
| | **User** | Specifies the user name of the account. |
| | **Password** | Specifies the password for the account. |
| **Set Indexer Administrative User Info?** | | Specifies you want to set an administrative indexer user.<br>Type the administrator user name and password on the Indexer machine. |

For a full description of indexer installation, see *Install the Indexer (TcFTSIndexer)*.

**Return to components list**

### Indexing Engine

| Parameter | Description |
|---|---|
| **Indexing Engine Settings** | Specifies the user account under which the FMS server cache (FSC) service runs. Choose one of the following options: |
| **User** | Type the user name for the Solr administrator. These credentials must match the Indexer and the Active Content Structure Translator (if used). |
| **Password** | Type the password for the Solr administrator account. These credentials must match the Indexer and the Active Content Structure Translator (if used). |
| **Indexing Engine Service Settings** | Settings for the Indexing Engine Service. |
| **Install Indexing Engine as a Service?** | Select this check box ☑ if you want to install the Indexing Engine as a service.<br>If you clear this check box, you must start the Indexing Engine manually after deployment on the Indexing Engine machine. |
| **User** | Type the operating system user name and password on the Indexing Engine machine.<br>If the Indexing Engine machine is a Windows machine, include the domain name (**domain\user**). |
| **Password** | Type the password for the user account under which the Indexing Engine service runs. |

For a full description of indexing engine installation, see *Install Indexing Engine (Solr)*.

Return to components list

### Licensing server

| Parameter | Description |
|---|---|
| **Teamcenter Licensing Port** | Specifies the port used by the license server. |

Return to components list

### Microservice Node

| Parameter | Description |
|---|---|
| *Windows machines* | |
| **Install Process Manager as a Windows service** | Select this check box ☑ if you want to install the Teamcenter Process Manager as a service.<br>If you clear this check box, you must start the Teamcenter Process Manager manually after deployment on the Microservice Node machine. |

| Parameter | Description |
| --- | --- |
| **Windows Service Name** | Specifies a name for the service. This name will be displayed in the Services panel in the Windows Control Panel. |
| **Microservice Node Type** | Specifies the type of microservice node:<br>**Master** — The master microservice node in the Teamcenter environment. Exactly one master microservice node is required in an environment. A master node must be configured before worker nodes are configured.<br>**Worker** — A worker microservice node in the Teamcenter environment.<br>You can add worker Microservice Node components as needed. |
| *Linux machines* | |
| **Instances** | Enter the number of service dispatcher instances to run on the node. |
| **Service Dispatcher Endpoint URL** | Enter the ingress URL for the service dispatcher. |
| **Keystore Password** and **Confirm Keystore Password** | Enter a password to be used for generating the **.p12** files that contain keys for signing and validating authentication tokens. The tokens identify the logged-on user.<br>Record and store the password securely for potential use, should you want to open and edit the keys. |
| **File Repository Storage Location** | Specifies the path to the storage location for the file repository to be used by the Active Workspace Gateway, for example, **c:\tc\file_repository**.<br>On Linux machines, the path to the storage location must exist on the current host. On Windows hosts, Deployment Center creates the directory if it does not exist.<br>If you install multiple instances of the File Repository microservice, all instances must reference the same physical storage location.<br>Active Workspace uses a file repository microservice. To configure that service for deployment on a Linux host, parameter values Deploying User UID, Deploying User GID, and File Repository Storage Location values are required. Values entered for the master microservice node must be valid on all worker nodes. |
| **Services** | In the **Services** list, review the quantity of instances for each service.<br>Typically, Teamcenter microservices are multi-threaded, so only one instance of the microservice is needed on a server.<br>When the environment includes multiple microservice nodes, you may want to run only a subset of microservices on a given node. In that case, for |

| Parameter | Description |
| --- | --- |
| | microservices that you do not want to install on the node, set the instance value to zero. |

Remaining **Microservice Node** parameters vary depending on your selections. For a full description of microservice deployment, see *Microservices and the microservice framework*.

Return to components list

### Server Manager

| Parameter | Description |
| --- | --- |
| **Server Pool ID** | Specifies a name for the server pool. |
| **Startup Mode** | Select one of the following:<br>• **Service/Daemon**<br>Specifies that you want to run the server manager as a Windows service. This is the default mode.<br>• **Command Line**<br>Specifies you want to run the server manager manually from a command line. |

If you want to set the **Config ID** value, which is included in the **Teamcenter Server Manager Service** name, click **Show All Parameters**.

For a complete description of server manager installation, see *Install the server manager*.

Return to components list

### Server Manager Cluster Configuration

| Parameter | Description |
| --- | --- |
| **Server Manager Cluster ID** | Type a name for the server manager cluster.<br>To balance the sessions load among multiple server managers, each server manager must have the same **Cluster ID** (that is, use the same server manager database). The **Cluster ID** value is stored in the **MANAGER_CLUSTER_ID** property in the *TC_ROOT***pool_manager\serverPool***database-name**.properties** file. |
| **Server Manager Database** | |

| Parameter | Description |
| --- | --- |
| **Creation Settings** | |
| **Create new database for the Server Manager Cluster** | Choose this option to create a new database user and database for the server manager. In the **Database Path** box, type the directory in which to create the Teamcenter database on the database server.<br>Be prepared to enter database system credentials for the new database. |
| **Use an existing database for the Server Manager Cluster** | Choose this option if you want to use an existing database user and database for the server manager. Your database administrator must create the database user and database before you proceed. |
| **Server Manager Database Settings** | |
| *Oracle databases* | |
| **Database Server** | Select **Oracle**. |
| **Port** | Specifies the number of the port on which the Oracle server listens. The port number was determined when the Oracle server was installed. |
| **Service** | Specifies the name of the service for the Oracle instance. The service name was determined when the Oracle server was installed. |
| **User** | Specifies the database user name. |
| **Password** | Specifies the database password. |
| *Microsoft SQL Server databases* | |
| **Port** | If you connect to Microsoft SQL Server using a specific port, select this option and enter the port number you specified when you installed MS SQL Server. |
| **Instance** | If you connect to Microsoft SQL Server using a named instance, select this option and enter the instance name you defined when you installed MS SQL Server. |
| **Database Name** | Specifies the name of the MS SQL Server database. The database name was determined when database was created. |
| **Collation** | Specifies the collation used by the Teamcenter database on the Microsoft SQL Server server. *Collation* defines the alphabet or language whose rules are applied when data is sorted or compared. |
| **User** | Specifies the database user name. |
| **Password** | Specifies the password for the database user. |

| Parameter | Description |
|---|---|
| **Database System User Credentials** | Database system credentials. These parameters are enabled if you chose the first option under **Server Manager Database Creation Settings**: |
| **System User** | Specifies the user name of the database system administrator account. For Oracle databases, the default value is **system**, for Microsoft SQL Server databases, the default value is **sa**. |
| **Password** | Specifies the password for the database system administrator account.<br>The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab. In addition, the password must not contain any of the following characters:<br>**! # @ $ % = & ' " ^ : ; . _ < > ( ) { }** |

For a complete description of server manager installation, see *Install the server manager*.

Return to components list

**Teamcenter Vault**

In the **Teamcenter Vault Service Name** box, specify a name for the vault service. Then, choose a **Vault Configuration** type and enter configuration parameters according to the type you select.

Two vault configuration types are supported:

| | |
|---|---|
| **Open Source** | Specifies the open source edition of HashiCorp Vault. To use this option, you must supply an available port and cluster port on the Teamcenter Vault machine. Deployment Center installs the vault. |
| **Enterprise Vault** | Specifies the enterprise edition of HashiCorp Vault. To use this option, you must install this edition of HashiCorp Vault *before* you install Teamcenter. You must also create a namespace and supply the URL to the vault and a one-time token to configure the vault. |

The required Teamcenter Vault parameters vary according to the **Vault Configuration** type you select:

| Vault Configuration Type | Description |
|---|---|
| **Open Source** | Specifies the open source edition of HashiCorp Vault. |
| **Port** | Specifies the port used by the Vault service. Make sure the port you specify is available on the Vault machine.<br>The default value is **8200**. |
| **Cluster Port** | Specifies the port used by the Vault service for cluster server requests. Make sure this port is *not* in use by another service. |

| Vault Configuration Type | Description |
|---|---|
| | The default value is **8201**. |
| **Teamcenter Vault URL** | Specifies the URL to the Teamcenter vault (HashiCorp open source edition vault). |
| **Enterprise Vault** | Specifies the enterprise edition of HashiCorp Vault. |
| **Enterprise Vault URL** | Specifies the URL to the HashiCorp Enterprise Edition Vault. |
| **Enterprise Vault Namespace** | Specifies the namespace in which the secrets engine and authentication are enabled. |
| **Enterprise Vault Authentication Namespace (Optional)** | Specifies then namespace where authentication is enabled in case the authentication namespace differs from the enterprise vault namespace. |
| **Enterprise Vault Authentication Path** | Specifies the patch to the vault authentication namespace. |

For more information about Teamcenter Vault installation and configuation, see *Teamcenter Administration*.

Return to components list

**Teamcenter Web Tier (Java EE)***(Java EE architectures only)*

| Value | Description |
|---|---|
| **Port** | Specifies the port to use to connect to the web tier. |
| **Teamcenter 4-tier URL** | Specifies the URL to the Teamcenter web tier application. This value is constructed from other parameters and is not directly editable. It has the following form:<br><br>*protocol*:**//***machine*:*port*/*application-name*<br><br>For example:<br><br>**https://myCorp:7001/tc** |
| **Teamcenter Application Name** | Specifies a name for the Teamcenter web tier web application. The default value is **tc**. |
| **Web App Server Machine Name** | Specifies the name of the machine that runs the Java EE web application server. This the machine on which you deploy the Java EE web tier WAR file (typically **tc.war**). |
| **JMX RMI Port** | Specifies the JMX RMI port number for the web server. For example, type **8088** for the default server manager port or **8089** for the default web tier port. |

| Value | Description |
|---|---|
| **Teamcenter Connection Name** | Specifies a name for the web tier connection. |
| **Tag** | Specifies a tag for the environment that can be used to filter the list of TCCS environments during logon. |

For a complete description of .Java EE web tier installation, see **Install the Java EE web tier**.

**Return to components list**

**Teamcenter Web Tier (.Net)***(.NET architectures only)*

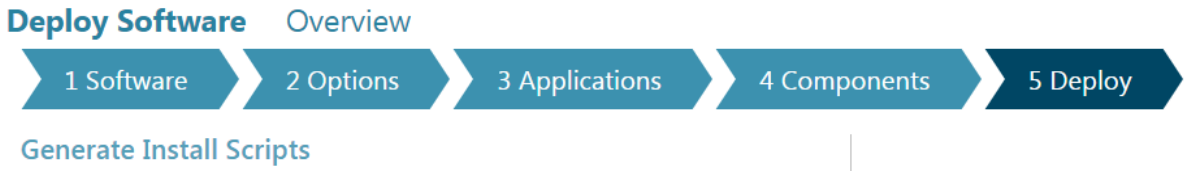
| Value | Description |
|---|---|
| **Protocol** | Specifies the protocol to use to connect to the web tier (**http** or **https**). |
| **Teamcenter 4-tier URL** | Specifies the URL to the Teamcenter web tier application. This value is constructed from other parameters and is not directly editable. It has the following form:<br><br>*protocol***://***machine***:***port***/***application-name*<br><br>For example:<br><br>**https://myCorp:7001/tc** |
| **Virtual Directory Name** | Specifies the IIS virtual directory name for Teamcenter .NET web tier deployment. The default value is **tc**. |
| **Teamcenter Connection Name** | Specifies a name for the web tier connection. |
| **Tag** | Specifies a tag for the environment that can be used to filter the list of TCCS environments during logon. |

For a complete description of .NET web tier installation, see **Install the .NET web tier**.

**Return to components list**

When all components are fully configured (showing a value of **100%** in the **COMPLETE** column), proceed to the **Deploy** tab.

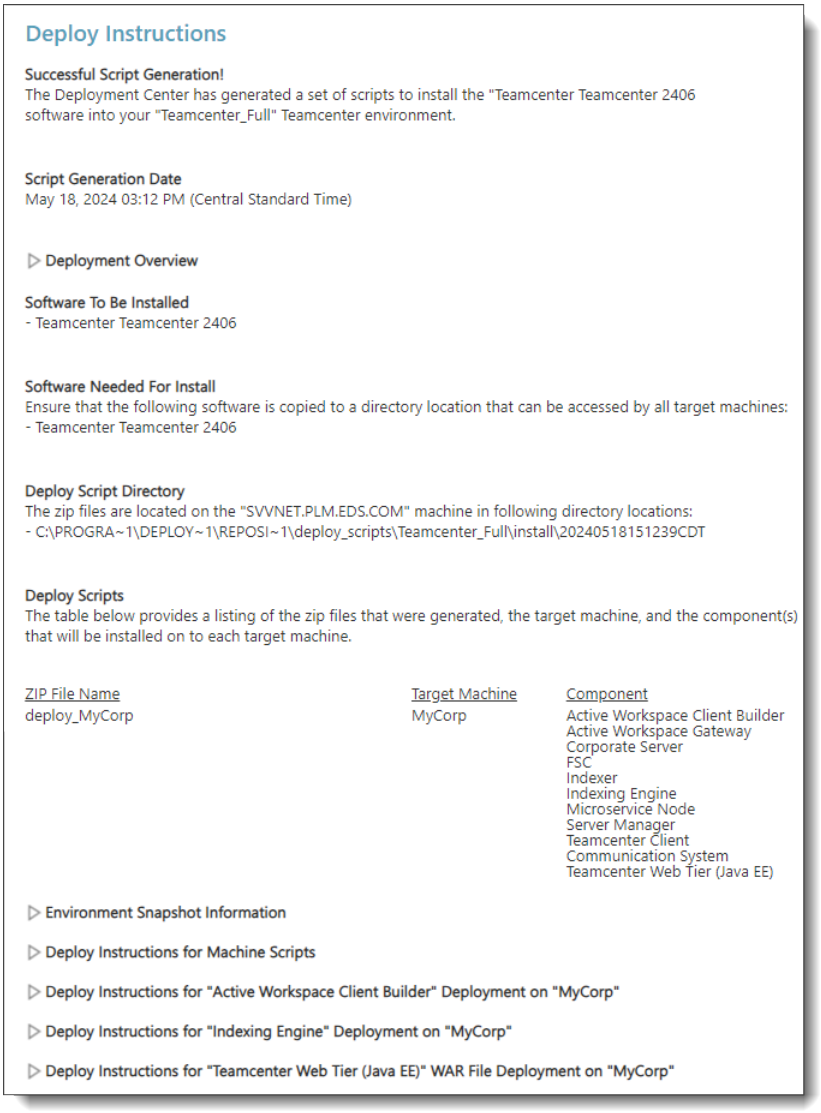
## Deploy the environment




In this tab, generate deployment scripts for each machine in your environment. These scripts install the software, applications, and components on to each target machine in your environment.

1. To generate deployment scripts, click **Generate Install Scripts**.

   Deployment Center generates installation scripts, and reports information about the scripts in the right panel.



The **Deploy Instructions** contain the following sections:

- **Script Generation Date** displays the time stamp for the local date and time of script generation.
- **Deployment Overview** describes the deployment covered by the scripts.
- **Software To Be Installed** lists the software required to deploy the components.
- **Software Needed For Install** lists software that is already installed on the machine but is still needed for this process to deploy other components.
- **Deploy Script Directory** displays the path to the location of the ZIP files containing the generated scripts. Go to the ZIP file directory and check for one or more ZIP files corresponding to the machines in your Teamcenter environment. Look for the *Deploy_Instructions.html* file, which contains the same information and instructions that you reviewed in the report.
- **Deploy Scripts** displays the ZIP files that were generated for each server along with the associated component names. Each ZIP file contains the installation scripts for a single server.

  If all components are to be installed on the same machine, there is only one ZIP file. The ZIP file name ends with the target machine name where you run the script. For example, if the ZIP file is named *deploy_MyCorp1.zip*, it runs on the **MyCorp1** machine. Run an installation script only on its designated machine.

2. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

   For more information about running deployment scripts, see *Run the deployment scripts*.

   If you encounter any issues during the deployment, see *Troubleshooting deployment*.

if you want to replicate an environment, you can export the configuration of an existing environment and then reuse its configuration to create another environment using the quick deployment procedure.

# 11. Complete the Teamcenter server installation

## Run the postinstallation tasks script (Linux systems)

On Linux systems, if you installed the corporate server without root privileges, a user with root privileges must run the root postinstallation tasks scripts. These scripts register daemons and perform other installation actions that require root privileges.

Run all scripts in the *TC_ROOT*/**install** directory that have names of the following form:

**root_post_tasks***ID*.**ksh**

Replace *ID* with the unique part of each script name.

## Start database daemons

### Starting database daemons on Windows

If you select the **Database Daemon** component during Teamcenter installation, Deployment Center configures the database daemons to start automatically as Windows services on Windows machines. After installation, you can find these services in the **Services** dialog box in the Windows Control Panel:

**Teamcenter Action Manager Service**
**Teamcenter Subscription Manager Service**
**Teamcenter Task Monitor Service**
**Teamcenter Tesselation Manager Service**
**Teamcenter Shared Metadata Cache Service**
**Teamcenter 4GD Change Detection Service**
**Teamcenter Revision Configuration Accelerator Service**
**Teamcenter Read Expression Manager Service**[1]
**Teamcenter Workflow Remote Inbox Sync Service**

If the services do not start automatically, see the available **troubleshooting solutions**.

Each service behaves as follows:

1. After the services are started, a program runs in *TC_ROOT***bin** named **tc_service.exe**.

   Windows displays **tc_service.exe** in the task manager. If you do not see this process, either your registry entry for that service is corrupted (specifically the path to the image) or the file is not on the system.

1 This service is installed by default and is not selectable in Deployment Center

2. The **tc_service.exe** program identifies the service that launched it by examining the service name. It expects the service name to contain either **actionmgrd**, **subscripmgrd**, **task_monitor**, or **tess_server**. The default service names for Teamcenter are **tc_actionmgrd**, **tc_subscripmgrd**, **tc_taskmonitor**, and **tc_tess_server**. These services are defined in **\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**.

3. The **tc_service.exe** program assembles a .bat file name by prefixing the service name with **run_** and appending the extension of .bat. For example, the **tc_actionmgrd** service has the file name **run_tc_actionmgrd.bat**.

4. The **tc_service.exe** program calls the .bat file (created by the setup program during configuration and placed in the **\bin** directory of the Teamcenter application root directory).

5. The task manager displays the process, for example, **actionmgrd.exe**.

   If the process is not displayed in the task manager, either the service name is not one of the three supported names, the .bat file for the process does not exist, or the process executable is missing.

6. The **Services** dialog box is updated to **Started**.

### Starting database daemons on Linux

You can start Teamcenter database daemons manually by executing the following startup files.

| Database daemon | Daemon startup script name |
|---|---|
| Action Manager Service | **rc.ugs.actionmgrd** |
| Subscription Manager Service | **rc.ugs.subscriptionmgrd** |
| Teamcenter Task Monitor Service | **rc.ugs.task_monitor** |
| Tesselation Manager Service | **rc.ugs.tess_server** |
| Teamcenter Shared Metadata Cache Service | **rc.ugs.shared_metadata** |
| Teamcenter 4GD Change Detection Service | **rc.ugs.4gd_change_detection_service** |
| Teamcenter Revision Configuration Accelerator Service | **rc.ugs.revision_config_accelerator** |
| Teamcenter Read Expression Manager Service[2] | **rc.ugs.am_read_expression_manager** |
| Teamcenter Workflow Remote Inbox Sync Service | **rc.ugs.schmgtwfd** |

Deployment Center places these startup files in the *TC_ROOT***/bin** directory.

2 This service is installed by default and is not selectable in Deployment Center

# Install database triggers manually

The **TcFTSIndexer** process requires database triggers that enable database access for the Indexer to detect additions, modifications, and deletions to the database when performing run-time (synchronous) indexing.

## Install database triggers in Oracle

1. To grant the **create trigger** privilege to the Oracle user that owns the Teamcenter database, perform the following steps:

   a. Open a command prompt.

   b. Type:

      **sqlplus system/***password*

   c. Type:

      **grant Create trigger to** *Tc-Oracle-user* **identified by** *password***;**

   d. Type:

      **exit**

2. Create the trigger:

   a. In the command prompt, type:

      **sqlplus** *Tc-Oracle-user***/***password*

   b. Type:

      **@***Teamcenter-installation-media***\tc\install\sitecons\sitecons_install_triggers_oracle.sql**

## Install database triggers in Microsoft SQL Server

1. Open Microsoft SQL Server Management Studio.

2. Complete the **Connect to Server** dialog box:

   a. In the **Server name** box, select the host on which Microsoft SQL Server is installed.

   b. In the **Authentication** box, select **SQL Server Authentication**.

c. In the **Login** box, type the database administrative user name.

d. In the **Password** box, type the database administrative user password.

e. Click **Connect**.

3. In the **Object Explorer** panel of the **Microsoft SQL Server Management Studio** dialog box, expand the **Databases** tree and select the Teamcenter database name, for example, **tc**.

4. From the menu bar, choose **File→Open→File**.

5. In the **Open File** dialog box, navigate to the software kit for the Teamcenter release.

   In the *Tc-software-path***tc\install\sitecons** directory, select **sitecons_install_triggers_mssql.sql**.

   Microsoft SQL Server Management Studio opens the selected file.

6. Click **Query→Execute**.

   The query installs the database triggers.

7. Verify that the query completed with no errors.

8. Close the Microsoft SQL Server Management Studio.

### Installing database triggers from the command line

If Microsoft SQL Server Management Studio is not installed on your host, you can install the database triggers from a command line. Type the following command in a Windows command prompt:

```
sqlcmd -H host -d database -U user -P password -i
path\sitecons_install_tables_and_triggers_mssql.sql
```

Replace:

- *host* with the database server host name.
- *database* with the Teamcenter database name.
- *user* with the database user name.
- *password* with the database user password.
- *path* with the path to the **sitecons_install_triggers_mssql.sql** file.

For example:

```
sqlcmd -H myhost -d TcDB -U dbUser -P dbPassword -i
      C:\software\tc\install\sitecons\sitecons_install_triggers_mssql.sql
```

To verify the triggers installed successfully, log into Microsoft SQL Server and type the following commands in an SQL prompt:

```
1> Select name,is_disabled from sys.triggers2> Go
```

If the installation succeeds, Microsoft SQL Server displays a table similar to the following showing that the database triggers are not disabled:

```
name                                is_disabled
------------------------------------------------------------------
fast_sync_add_trigger                        0
fast_sync_delete_trigger                     0
(2 rows affected)
```

# 12. Installing distributable components

## Install the server manager

1. Log on to Deployment Center and select your environment.

2. In the **Components** tab, select the **Server Manager** component.

3. Enter values for the machine on which you install the server manager:

   - **Single box**

     If your environment is a **single box** environment, the **Machine Name**, **OS**, and **Teamcenter Installation Path** values are inherited from the first component you configured in your environment. Changing these values will change them for other components in your environment.

   - **Distributed**

     If your environment is a **distributed** environment, type the **Machine Name**, **OS**, and **Teamcenter Installation Path** for the machine on which you install the server manager.

4. Enter required values to configure the server manager:

   | | | |
   |---|---|---|
   | **Server Pool ID** | Specifies a unique ID for this pool of server processes. | |
   | **Startup Mode** | Specifies how you want to start the server manager: | |
   | | **Service/ Daemon** | Specifies you want to run the server manager as a service (a system service on Windows or a daemon on Linux). |
   | | **Command Line** | Specifies you want to run the server manager manually from a command line. |

   If you want to specify additional settings for the Indexing Engine, click **Show all parameters**  .

5. Proceed to configuring the **Server Manager Cluster Configuration** component, which Deployment Center automatically selects for configuring next.

   As with the **Server Manager** component, enter values for **Machine Name**, **OS**, and **Teamcenter Installation Path** as appropriate for your environment type (single box or distributed).

6. Enter the required values to configure the server manager cluster:

**Server Manager Cluster ID**

Type a name for the server manager cluster.

To balance the sessions load among multiple server managers, each server manager must have the same **Cluster ID** (that is, use the same server manager database). The **Cluster ID** value is stored in the **MANAGER_CLUSTER_ID** property in the *TC_ROOT***pool_manager\serverPool***database-name**.**properties** file.

**Server Manager Database Creation Settings**

Choose one of the following options:

- **Create new database for the Server Manager Cluster**

  Choose this option to create a new database user and database for the server manager. In the **Database Path** box, type the directory in which to create the Teamcenter database on the database server.

  Be prepared to enter database system credentials for the new database.

- **Use an existing database for the Server Manager Cluster**

  Choose this option if you want to use an existing database user and database for the server manager. Your database administrator must create the database user and database before you proceed.

**Server Manager Database Settings**

Select the database vendor (**Oracle** or **MSSQL Server**), then enter the appropriate database configuration values:

**Table 3-4. Oracle database server values**

| Value | Description |
|---|---|
| **Port** | Specifies the number of the port on which the Oracle server listens. The port number was determined when the Oracle server was installed. |
| **Service** | Specifies the name of the service for the Oracle instance. The service name was determined when the Oracle server was installed. |
| **User** | Specifies the database user name. |
| **Password** | Specifies the database password. |

**Table 3-5. Microsoft SQL Server database server values**

| Value | Description |
|---|---|
| **Instance** | If you connect to Microsoft SQL Server using a named instance, select this option and enter the instance name you defined when you installed MS SQL Server. |
| **Port** | If you connect to Microsoft SQL Server using a specific port, select this option and enter the port number you specified when you installed MS SQL Server. |

| Value | Description |
| --- | --- |
| **Database Name** | Specifies the name of the MS SQL Server database. The database name was determined when database was created. |
| **Collation** | Specifies the collation used by the Teamcenter database on the Microsoft SQL Server server. *Collation* defines the alphabet or language whose rules are applied when data is sorted or compared. |
| **User** | Specifies the database user name. |
| **Password** | Specifies the password for the database user. |

**Database System User Credentials**

In the **System User** box, type the user name of the database server system administrator account. For Oracle databases, the default value is **system**. For Microsoft SQL Server databases, the default value is **sa**.

In the **Password** box, type the password for the database server system administrator account.

> Caution:
>
> The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab. In addition, the password must not contain any of the following characters:
>
> **! # @ $ % = & ' " ^ : ; . _ < > ( ) { }**

7. Click **Save Component Settings** to submit the server manager cluster configuration values.

8. Complete configuration of any remaining components.

9. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

10. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see the *Deployment Center — Usage*.

If you experience connection delays during server manager startup, see the **available troubleshooting solutions**.

### Java EE configuration files

You can install multiple server manager services (on Windows systems) or daemons (on Linux systems) on the same machine. Each server manager service has its own configuration directory:

**Windows:** *TC_ROOT***pool_manager\confs***config-name**

**Linux:** *TC_ROOT*/**pool_manager/confs/***config-name*

where *config-name* is the name of the server manager.

The server manager configuration directory contains configuration files, log files, and server manager scripts. These include the following.

| File/Directory | Description |
|---|---|
| **mgrstart** | Script that launches the server manager in console mode. |
| **mgrstop** | Script that stops the server manager when started from a command line.<br>On Linux systems, if you run the server manager as a daemon, stop the service using the **rc.tc.mgr_***config* script.<br>On Windows systems, if you run the server manager as a Windows service, stop the service using the Windows services manager.<br>You can also stop the server manager using the Teamcenter Management Console. |
| **mgr.output** | If you run the server manager as a service (on Windows systems) or a daemon (on Linux systems), this file contains all output from the server manager.<br>This file is *not* used if you run the server manager from the command line. |
| **logs** | Directory that contains all server manager log files. |

If you run the server manager as a Windows service or a Linux daemon, the server manager starts automatically.

# Installing Teamcenter microservices

## Microservices and the microservice framework

Various Teamcenter solutions and applications include microservices as part of their deployment. For example:

- Active Workspace requires TcGQL and File Repository microservices.

  The File Repository provides centralized temporary storage for web client content accessed through the web client gateway. This storage gives other microservices an alternative to the File Management System (FMS).

- The Classification and Requirements Manager applications each have their own required microservices.

- The Product Configurator application can optionally employ its application-specific microservice to achieve better performance.

The microservice framework enables microservices to run seamlessly across diverse platforms.

To install the microservice framework and the microservices that run on it, you must configure and deploy a microservice node. If the server hardware has sufficient capacity, you can deploy a microservice node on the same hardware as a Teamcenter pool manager.

To increase capacity and provide failover, the microservice framework can include multiple nodes. For Linux deployments, a single node configuration is reused by the Docker swarm or the Kubernetes cluster. For Windows deployments, you can add and configure worker microservice nodes in addition to a master microservice node.

All microservice nodes in a Teamcenter environment must be hosted on servers of a single operating system type. The following table compares the characteristics of microservice nodes hosted on Linux and Windows.

| | **Linux 64-bit** | **Windows 64-bit** |
|---|---|---|
| **Prerequisite third-party software** | On the microservice node:<br>• Mirantis Container Runtime (formerly Docker Engine - Enterprise)<br>• Kubernetes (only if deploying into a Kubernetes environment)<br>In a location accessible from the microservice node:<br>• A container registry | None |
| **Management of microservice framework and application microservices** | Docker Swarm or Kubernetes starts, stops, restarts, and scales all Teamcenter microservices running as containers in a way that best utilizes resources. | On Windows, each microservice framework node includes a Teamcenter process manager to handle the microservices on that node. |

## Microservice framework constituents

The microservice framework has the following constituents:

| | |
|---|---|
| **Service Registry** | Maintains a list of running microservice instances across all nodes. |
| **Service Dispatcher** | Receives microservice requests from a Teamcenter client, queries the service registry to find an instance of the requested microservice, and then routes the request to an instance of the microservice. |
| **Microproxy** | Forwards web tier application requests to the service dispatcher. |
| **Teamcenter Process Manager** (Windows hosts) | Starts the microservices and Active Workspace Gateway on the node (Windows hosts). |

| | |
|---|---|
| **Microservice Parameter Store** (MPS) | Manages logging levels for microservices. |
| **File Repository** | Manages files for web client and microservices. |

## Install microservices on Linux

### Deploy MCR (Docker) on microservice node hosts

Mirantis Container Runtime (MCR, formerly Docker Engine - Enterprise) is a prerequisite for microservice nodes on Linux hosts. For certified versions of Linux and MCR (Docker) software, refer to the *Hardware and Software Certifications* knowledge base article on Support Center.

**Install and configure MCR**
**Working with Docker containers in Docker Swarm**
**Docker troubleshooting**

### Install and configure MCR

1. Ensure the following ports are open to traffic to and from each microservice node host:

| Port | Traffic type |
|---|---|
| TCP port 2377 | Cluster management communications |
| TCP and UDP port 7946 | Communication among nodes |
| UDP port 4789 | Overlay network traffic |

2. Install MCR.

3. Configure MCR to restart on system boot.

4. Configure IPv4 forwarding.

   IP forwarding must be enabled for successful communication between Docker containers and the host machine. MCR installation alters the Linux iptables to allow forwarding of packets between the host and bridge networks when such forwarding is enabled. See MCR (Docker) documentation for information on how to partially restrict forwarding (based on IP addresses) for tighter security.

   IP forwarding is controlled by Linux kernel parameters such as **net.ipv4.ip_forward** and **net.ipv4.conf.all.forwarding**, depending on the distribution and version of Linux. To check the current setting, you can use the command **sysctl net.ipv4.ip_forward** (sudo or root access is required). The value **0** disables forwarding; the value **1** enables forwarding.

   a. To temporarily enable IP forwarding for testing, run the following command:

      **sysctl -w net.ipv4.ip_forward=1**

b. To restart MCR, run the following command:

**systemctl restart docker**

c. To preserve this setting across a machine reboot, edit the file **/etc/sysctl.conf** and set **net.ipv4.ip_forward** to **1**.

## Working with Docker containers in Docker Swarm

Use common Docker commands to control the Docker Swarm environment and monitor container status. Additionally, many open source tools, such as Portainer, are available to help manage a Docker Swarm.

### Commonly used Docker commands

| To do this | Run this command |
|---|---|
| List the Docker container stacks. | **docker stack ls** |
| List the services currently running. | **docker service ls** |
| Display the last five lines that were output by a particular service. | **docker service logs -f --no-task-ids --tail 5** *service_id*<br>A *service_id* has the form *stackname_servicename*. |
| List the nodes in a swarm. | **docker node ls** |
| List the images registered in the container registry on a node. | **docker image ls** |

### Managing containers with Portainer

Portainer is an open source product that provides a web-based UI to easily manage Docker swarms, services, and containers. You can use Portainer to do the following:

- View Docker container log files.
- View the Docker applications (stacks) that have been started.
- View the status and location of running services.
- Manage the nodes in a swarm and temporarily adjust scaling of services across the swarm.

## Docker troubleshooting

### What do I do when I receive the error `Cannot connect to the Docker daemon`?

1. To check whether **dockerd** is running, run

**ps -eaf | grep dockerd**

2. Perform remedial steps depending on the result from Step 1.

| If dockerd is | Then do this |
|---|---|
| Not running | Restart Docker, and configure **dockerd** to restart on the next boot:<br><br>`sudo systemct1 start docker`<br>`sudo systemct1 enable docker` |
| Running | The user is likely not a member of the Docker Linux group.<br><br>Add the user to the group. Ignore any error output from **groupadd**.<br><br>`sudo groupadd docker`<br>`sudo usermod -aG docker $USER` |

For more debugging information, refer to configuring the Docker daemon in the documentation at **https://docs.docker.com**.

**What do I do if a Docker command does not behave as expected?**

If the Docker command does not behave as expected, add the **-debug** option, run the command again, and review the log for issues.

> Example:
>
> You run the command **docker deploy -f mystack.yml mystack** and it does not behave as expected.
>
> To enable logging, insert **-debug** after **docker**:
>
> **docker -debug deploy -f mystack.yml mystack**.

**How do I view logs from the Docker daemon?**

To view logs from the Docker daemon, open a new shell and enter the following:

```
sudo journalctl -fu docker.service
```

This tails the log files and keeps outputting new log commands until the command prompt is closed or you enter **Ctrl-C**.

**Where can I get help with more complicated environments?**

For help with more complicated environments and networking when microservice nodes are on Linux hosts, see the Docker engine swarm mode documentation at **https://docs.docker.com/**.

### Deploy a Docker container registry

For deployments of the microservice framework and microservices on Linux hosts, microservice container images are stored in a container registry. If you do not already have a container registry in your infrastructure, you can use the following procedure to deploy a Docker container registry.

For detailed documentation on Docker Registry, see **https://docs.docker.com/registry/**.

**Prerequisites**

- Mirantis Container Runtime (formerly Docker Engine - Enterprise, hereafter referred to as MCR) must be installed on both the machine that is used to initially fetch the Docker Registry container image and on the machine that will host the Docker Registry. For instructions on installing MRC, refer to **Deploy MCR (Docker) on microservice node hosts**.

- If the microservice framework is to be deployed on a Kubernetes cluster, Kubernetes must be installed on the Docker Registry machine in addition to MCR.

- For a secure production environment, a PKI certificate and keys generated for the server hosting the Docker Registry must be available. It is a good practice to obtain certificates from a Certificate Authority.

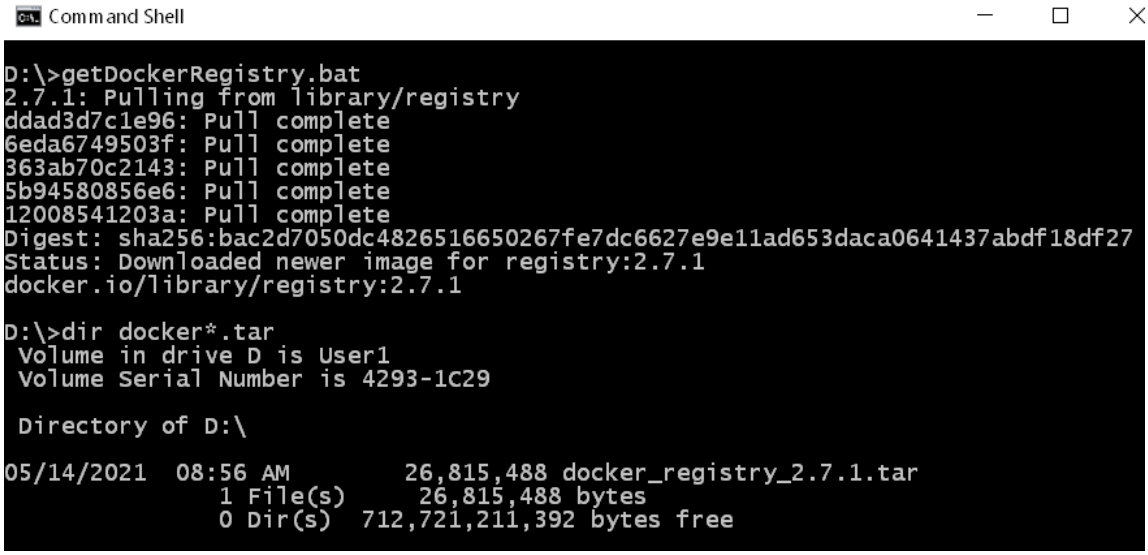
1. Prepare to fetch the Docker Registry container image. On a machine with internet access and with MCR installed, extract the microservice framework Linux kit.

   If the machine is not the intended Docker Registry server, its host operating system can be either Linux or Windows.

2. Copy the **/additional_applications/docker_registry.zip\additional_applications\docker_registry.zip** file from the Teamcenter software kit to a local directory, and then unzip its contents.

3. In the directory that contains the unzipped file contents, run the appropriate script to fetch the tested version of the Docker Registry container image.

| | |
|---|---|
| Linux | **getDockerRegistry.sh** |
| Windows | **getDockerRegistry.bat** |




4. As needed, move the fetched **.tar** file to the machine that will run the Docker Registry service. Load the image.

   `docker image load -i` *tar_file_name*

5. Deploy the registry for the planned microservice framework container manager type, either Docker Swarm or a Kubernetes cluster.

   a. Create the following directories:

      **/scratch/docker_registry/data**

      **/scratch/docker_registry/certs**

      **/scratch/docker_registry/auth**

      If you use different paths, update the YML or YAML configuration files in the corresponding subdirectories of the microservice framework kit:

      *kit***additional_applications\docker_registry\deploy***swarm** or *kubernetes*

   b. Add your certificate files to **/scratch/docker_registry/certs**.

      See **https://docs.docker.com/registry/deploying/#run-an-externally-accessible-registry**

   c. Restrict access.

      See **https://docs.docker.com/registry/deploying/#restricting-access**

d. Deploy the registry to a new stack. This new stack is unrelated to Teamcenter, Active Workspace, and the microservice framework.

| For this environment | Issue these commands |
|---|---|
| Docker Swarm | `docker stack deploy -c path/docker_registry.yml tcregistry` |
| Kubernetes cluster | `kubectl create namespace tcreg`<br>`kubectl apply -f path/docker_registy.yaml -n tcregistry` |

> Caution:
>
> Deploy to a unique cluster or stack separate from Teamcenter microservices. This protects the running registry if you delete the Teamcenter microservices Docker Swarm stack or Kubernetes cluster.

### Validate functionality of Docker Registry

After you deploy the registry, check to see that it is running.

1. Run the following command to list the registry contents.

   ```
   curl --cacert /scratch/docker_registry/certs/domain.crt https://vcl6006:5000/v2/_catalog
   ```

   The valid response shows an empty repository, as nothing has been pushed to it yet:

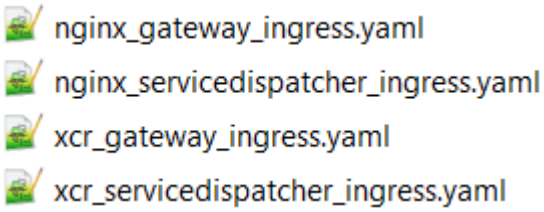
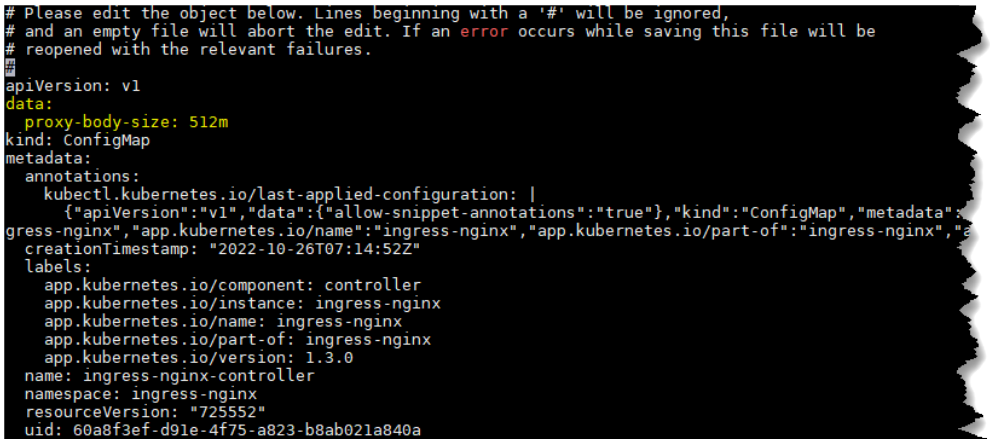
   ```
   {"repositories":[""]}
   ```

#### Install microservices on a Linux machine

Microservices can be installed on a Linux host that is either a member of a Docker swarm or managed by Kubernetes.

For installation in a Kubernetes environment, two prerequisites must be in place before configuring a microservice node and its microservices. These prerequisites are common tasks when setting up a Kubernetes environment. Resulting values are needed during configuration.

| Prerequisite for Kubernetes environment | Description |
|---|---|
| **Ingress controller** | Set up an ingress controller of your choosing. Configure rules for two routes:<br>• **/awc** which goes to the **gateway** service, port 3000<br>• **/sd** which goes to **service-dispatcher** service, port 9090 |

| Prerequisite for Kubernetes environment | Description |
|---|---|
| | Example ingress controller configurations are included in the Teamcenter installation kit for Linux. The location within the zipped kit is **tc*version*_lnx64.zip\tc\sample.zip\sample\kubernetes_templates\ingress**.<br><br>  <br><br><br>The ingress controller must also be configured to allow for attaching payloads of sufficient size in Active Workspace. The setting for this may vary depending upon which ingress controller is in use. Please refer to the documentation for your ingress controller.<br><br>Example:<br><br>For an nginx ingress controller, the solution is to define the following setting (highlighted in yellow) in the nginx config map:<br><br>  <br> |
| **PersistentVolume** | Set up a PersistentVolume and define a storageClassName for that volume. |

1. Download a compatible Teamcenter kit and place it in the Deployment Center software repository.
2. In Deployment Center, open or create an environment.
3. On the **Software** tab, add **Microservice Framework**.
4. On the **Applications** tab, add the applications that you want to install in the environment.

5. On the **Components** tab, specify values for the **Microservice Node** options.

| For this option | Do this |
|---|---|
| **Installation Path** | Enter the path to the Teamcenter installation root folder on the microservice node host machine. |
| **Machine Name** | Enter the fully qualified domain name of the microservice node host machine.<br>This machine name is used to construct the service dispatcher URL. |
| **OS** | Choose **lnx64** (Linux). |
| **Instances** | Enter the number of service dispatcher instances to run on the node. |
| **Protocol** | Choose the protocol to use for moving data between the Teamcenter web tier and the service dispatcher.<br>If you choose **https** (recommended), you must complete the configuration parameters in the **HTTPS Config** component.<br>To enable encryption using Kubernetes on Linux, deploy a service mesh such as Istio or Linkerd. Do not configure service dispatcher for HTTPS. |
| **Port** | Enter the port number for communication with the service dispatcher.<br>For Kubernetes, the valid port range is from 30000 to 32767. |
| **Additional Service Dispatcher URLs** | If additional cluster or swarm members will host a service dispatcher, click **Add URL** ⊕ and enter the URLs, including port values, to those service dispatchers. An example is **http://machine2:9090**.<br>Port and protocol values in the additional URLs must be the same as those specified in **Protocol** and **Port**. |
| **Keystore Password** and **Confirm Password** | Enter a password to be used for generating the **.p12** files that contain keys for signing and validating authentication tokens. The tokens identify the logged-on user.<br>Record and store the password securely for potential use, should you want to open and edit the keys. |
| **File Repository Storage Location** | Enter the path to the shared location for persistent file storage.<br>The path must be accessible by all microservice nodes. |
| **Deploying User UID** and **Deploying User GID** | Follow the instructions in Deployment Center to obtain and enter the UID and GID of the user who will deploy the file repository microservice.<br>Values entered must be valid on all swarm or cluster members that will run the file repository microservice.<br>For Kubernetes, the user cannot be **root**. |

6. Enter additional microservice parameter values as required. The parameters shown vary depending on which applications are selected for the environment.

7. In the **Services** list, review the quantity of instances for each service.

To increase capacity, increase the number of instances.

8. Save the component settings.

   Deployment Center copies the service dispatcher URLs to the Active Workspace Gateway and Web Tier components.

9. If you plan to use a load balancer for ingress to service dispatcher instances, go to the Active Workspace Gateway and Web Tier component panels, click **Show all parameters**  , and scroll to the **Microservice Node Connection(s)** table. Select **Override connection** and edit the table as needed to correctly specify the ingress URLs for the service dispatcher(s).

10. In the **Container Configuration** component, specify option values. The component appears only if **Microservice Node OS** is set to **lnx64**.

| For this option | Do this |
|---|---|
| **Container Registry URL** | Enter the machine name or IP address and port of the container registry. Do not enter a protocol. |
| **Container Repository Name** | Enter the name of the repository for Teamcenter microservices.<br>A repository is a logical grouping of container images within the registry. The repository name must exist in the container registry before you run the scripts generated by Deployment Center. The recommended name is **teamcenter**. |
| **Container Manager** | Choose one of two container manager types, **Docker Swarm** or **Kubernetes**.<br>For Kubernetes, specify the **Namespace**. A namespace is the unique name that identifies the group of Teamcenter resources interacting with each other in a Kubernetes cluster. The value you enter replaces placeholders in microservice **.yml** files. This is the same namespace described in the procedure **Deploy microservices in Kubernetes**. |

11. Complete the configuration of the environment and generate deployment scripts.

12. Ensure that Docker is installed on the microservice node host before you run its deployment script. Refer to **Deploy MCR (Docker) on microservice node hosts**.

13. Log on to the container registry before starting actual deployment.

    `docker login -u "`*user*`" -p "`*password*`" `*container_registry_URL*

14. Run the deployment scripts.

    Run the microservice node scripts before you run the web tier deployment script.

15. Depending on the container manager, follow the appropriate instructions to complete the installation and start the microservices:

| | |
|---|---|
| **Docker Swarm** | Start microservices in Docker Swarm. |
| **Kubernetes** | Deploy microservices in Kubernetes. |

### Add microservice instances for a Linux machine

To increase the capacity of heavily used microservices deployed to Linux hosts, you can add microservice instances via Deployment Center.

## Add microservice instances

1. In Deployment Center, on the **Components** tab for your environment, open the **Microservice Node**.

2. In the list of microservices, change values for the instances as desired.

3. Complete your environment configuration and follow the Deployment Center instructions for deploying the generated ZIP files onto the target machines.

4. Depending on the container manager in your environment, do one of the following:

   - Start microservices in Docker Swarm
   - Deploy microservices in Kubernetes

### Start microservices in Docker Swarm

When your microservice framework is deployed for Docker Swarm, use the following procedure to start Docker and then start microservices.

## Start Docker

To start Docker on a microservice framework node, run the following command:

```
docker swarm init
```

The output of the command is similar to the following:

```
Swarm initialized: current node (lccilqci5tpvy6xmsjlu8gap3) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-26h1be2gk2kozzecvgkw93smho5ueb7azn8uw1j2079
isc8b25-dfc8r1f6qhh50ev250tb4st9r 192.168.0.8:237
```

Tip:

If this is the master node and you intend to later join other servers to this swarm as workers, save the output command string for later use.

Once you have started Docker on a node, you can **join the node to a running swarm**.

### Deploy the microservice stack

During the installation of microservice nodes, one node must be configured. Microservice **.yml** files are copied to this node. These files define the microservice container parameters and are used to deploy the microservice containers. Once the stack of containers is deployed on this node, Docker manages the stack across the swarm, automatically deploying containers as needed on other servers that join the swarm.

1. Change to the Docker *installation-path***/container** directory.

2. Run the following commands to deploy a stack for the microservice framework service:

   ```
   docker stack deploy -c tc_eureka.yml myStackName
   docker stack deploy -c service_dispatcher.yml myStackName
   ```

3. Using the same command pattern and the same stack name, deploy all other **.yml** files in the directory.

#### Join a server to a Docker Swarm

Once a microservice node has started a Docker Swarm, you can join additional servers to the swarm as either *workers* or *managers*. Any number of servers can be added as workers. If the swarm includes multiple manager nodes, the manager nodes vote to determine which node is the controlling node. To ensure a decisive vote, the swarm must have an odd number of manager nodes.

1. **Start Docker** on the server.

2. Use the appropriate procedure to join the server to the swarm as a worker or as a manager.

| For this join mode | Do this |
| --- | --- |
| *Worker* | Run the Docker command that you saved from the output when the swarm was started.<br><br>`docker swarm join --token SWMTKN-1-26h1be2gk2kozzecvgkw93smho5ueb7azn8uw1j2079isc8b25-dfc8r1f6qhh50ev250tb4st9r 192.168.0.8:237` |

| For this join mode | Do this |
|---|---|
| | If a saved join token is not available, on the original node run the following command to request a token:<br><br>`docker swarm join-token` |
| *Manager* | a. Ensure that in Teamcenter Web Application Manager (**insweb**) you configure the Teamcenter WAR file to include the node host's URL in the **Context Parameters** value list for **MICROSERVICE_ADDRESS**.<br><br>b. On the original node, run the following command to request a manager token:<br><br>`$ docker swarm join-token manager`<br><br>The output of the command is similar to the following:<br><br>`To add a manager to this swarm, run the following command:`<br><br>`docker swarm join --token SWMTKN-1-26h1be2gk2kozzecvgkw93smho5ueb7azn8uw1j2079isc8b25-ct7cb2rwewvmffmi69c7gt1zn 192.168.0.8:2377`<br><br>c. Copy the command output and paste it to a command line on the machine you want to join to the swarm.<br><br>`docker swarm join --token SWMTKN-1-26h1be2gk2kozzecvgkw93smho5ueb7azn8uw1j2079isc8b25-ct7cb2rwewvmffmi69c7gt1zn 192.168.0.8:2377`<br><br>The output of the command is similar to the following:<br><br>`This node joined a swarm as a manager` |

## Deploy microservices in Kubernetes

If deploying the microservice framework into a Kubernetes environment, then after using Deployment Center to perform initial microservices configuration and installation, use either the following automated or manual procedure to finalize configuration and start microservices. A short list of commands useful for validating the microservice environment in Kubernetes follows the procedures.

**Deploy automatically**
**Deploy manually**
**Validate the microservice framework and microservices in a Kubernetes cluster**

Note:

By default, support is provided for **Project Calico** network policies. If your network policy solution is other than Project Calico, review the generated network security policy files (***_np.yml**) and create versions compatible with your network policy solution.

### Deploy automatically

The Deployment Center scripts deposit relevant shell scripts in the *TC_ROOT***/bin** folder on the microservice node machine.

**deploy_microservices.sh**
**redeploy_microservices.sh**
**undeploy_microservices.sh**

Run the relevant shell script.

```
./deploy_microservices.sh
```

The scripts are also present in the microservice framework kit within **additional_applications\microservice_management**.

### Deploy manually

1. Establish the namespace.

   a. Create a custom namespace.

      ```
      kubectl create namespace custom_namespace
      ```

      The namespace value should match the namespace value entered in the Container Configuration component for the environment in Deployment Center.

   b. Check your namespace.

      ```
      kubectl get namespace
      ```

   c. Change the context to the namespace.

      ```
      kubectl config set-context --current --namespace=custom_namespace
      ```

2. Create secrets and ConfigMaps.

   a. Change to the Kubernetes **scripts** directory.

```
cd TC_ROOT/container/kubernetes/setup/scripts
```

b. Run all scripts in the **scripts** directory.

> Caution:
>
> Address any errors before proceeding to the next step.

3. Deploy the microservice framework and microservices in a Kubernetes cluster.

a. Change to the Kubernetes setup directory.

```
cd TC_ROOT/container/kubernetes/setup
```

b. Run all the setup files (network policies, volumes, persistent volume claims, and persistent volume).

```
kubectl create -f .
```

c. Change to the Kubernetes deployment directory.

```
cd TC_ROOT/container/kubernetes/deployment
```

d. Create all the deployments (deploying microservices).

```
kubectl create -f .
```

### Validate the microservice framework and microservices in a Kubernetes cluster

`kubectl` is a utility to manage Kubernetes data.

| Validation step | Example command |
|---|---|
| Get the list of defined namespaces. | `kubectl get namespaces` |
| Get the list of running pods (containers). | `kubectl get pods -n=<namespace>` |
| Get the list of services running in the namespace along with the exposed port. | `kubectl get svc -n` *custom_namespace* |
| Check logs of the pods. | `kubectl logs` *pod_name* |
| Check environment variables of the pod. | `kubectl exec` *pod_name* `env` |
| Test microservice (in a web browser). | *IP_address*`:`*service_dispatcher_port*`/mps/health/checkhealth` |
| Get verbose information about a specified object. | `kubectl describe <object type> <id>` |

## Install microservices on Windows

### Install microservices on a Windows machine

1. Download the Teamcenter 2412 software kit for Windows and place it in the Deployment Center software repository.
2. In Deployment Center, open or create an environment.
3. On the **Software** tab, add the Teamcenter 2412 software.
4. On the **Applications** tab, add the applications that you want to install in the environment.
5. On the **Components** tab, specify values for the **Microservice Node** options.

| For this option | Do this |
|---|---|
| **Installation Path** | Enter the path to the Teamcenter installation root folder on the microservice node host machine. |
| **Machine Name** | Enter the fully qualified domain name of the microservice node host machine.<br>This machine name is used to construct the Service Dispatcher URL. |
| **OS** | Choose **wntx64** (Windows).<br>Check **Install Teamcenter Process Manager as a Windows service** to automatically start services when the server reboots. |
| **Microservice Node Type** | Choose one of two node types:<br>**Master**<br>The master microservice node in the Teamcenter environment. Exactly one master microservice node is required in an environment. A master node must be configured before worker nodes are configured.<br>**Worker**<br>A worker microservice node in the Teamcenter environment.<br>You can add worker Microservice Node components as needed. |
| **Keystore Password** and **Confirm Password** | Enter a password to be used for generating the **.p12** files that contain keys for signing and validating authentication tokens. The tokens identify the logged in user.<br>Record and keep secure the password for potential use should you want to open and edit the keys. |
| **Protocol** | Choose the protocol to use for moving data between the Teamcenter web tier and the Service Dispatcher. The default protocol is **http**. If the Teamcenter architecture type is **Java EE**, then you have the option of choosing **https**. |

| For this option | Do this | |
|---|---|---|
| | **For this web tier architecture** | **Do this** |
| | .NET | Choose **http**. |
| | Java EE | Choose either **http** or **https**.<br><br>If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| **Port** | As applicable, enter the port number for communication with the Service Dispatcher and the Service Registry.<br><br>Both the Service Dispatcher and the Service Registry are required on the master node. | |
| **Teamcenter Microservice URL** and **Service Registry URL** | As applicable, Deployment Center supplies these values as you complete the environment configuration. | |

6. Enter microservice parameter values as required. The parameters shown vary, depending on which applications are selected for the environment.

   > Example:
   >
   > Active Workspace uses a file repository microservice. To configure that service for deployment on a Windows host, in the parameter value for **File Repository Storage Location**, enter the path to the shared location for persistent file storage. The path must be accessible by all microservice nodes.

7. In the **Services** list, review the quantity of instances for each service.

   Typically, Teamcenter microservices are multi-threaded, so only one instance of the microservice is needed on a server.

   When the environment includes multiple microservice nodes, you may want to run only a subset of microservices on a given node. In that case, for microservices that you do not want to install on the node, set the instance value to zero.

8. Save the component settings.

   Deployment Center copies the generated service dispatcher URLs to the Active Workspace Gateway and Web Tier components.

9. If a microservice node does not include a service dispatcher instance, or if you plan to use a load balancer for ingress to service dispatcher instances, go to the Active Workspace Gateway and Web Tier component panels, click **Show all parameters** , and scroll to the **Microservice Node**

**Connection(s)** table. Select **Override connection** and edit the table as needed to correctly specify the ingress URLs for the service dispatcher(s). To remove a URL row, select the row and then click **Remove connection** .

10. Complete configuration of the environment and generate deployment scripts.

11. Run the deployment scripts.

    Run the microservice node scripts before you run the web tier deployment script.

12. If your environment uses the .NET architecture, on the web tier server machine where IIS is running, in **Application Pools** > **DefaultAppPool** or **Teamcenter App Pool** > **Advanced Settings**, set **Load User Profile** to **True**.

13. If your environment uses the Java EE architecture:

    a. Locate the WAR file (**tc.war**) in the **deployment** directory under the staging location you specified.

    b. Deploy the WAR file on a supported application server, as described in *Web Application Deployment* in the Teamcenter help.

14. Start the framework and services.

    Microservice processes are started by the Teamcenter Process Manager, which can be started as either a Windows service or from a startup file.

| | |
|---|---|
| **Windows service** | If you selected the **Install the Teamcenter Process Manager as a Windows service** option, then the Teamcenter Process Manager starts automatically with system startup.<br><br>The Teamcenter Process Manager appears in the Windows service list as **Teamcenter Process Manager** if POOL_ID is not defined, else **Teamcenter Process Manager <POOL_ID>**. |
| **Startup file** | Run the Teamcenter Process Manager startup file:<br><br>*TC_ROOT*`\process_manager\start_manager.bat` |

### Add microservices and microservice nodes Windows machine

To increase capacity of heavily used microservices deployed on Windows hosts, you can add microservice nodes and microservice instances via Deployment Center.

1. In Deployment Center, on the **Components** tab for your environment, select an existing microservice node component or add a new **Microservice Node** component.

2. Configure the node, including the microservices you want to run on the node, as described in **Install microservices on a Windows machine**.

3. Complete your environment configuration and follow the Deployment Center instructions for deploying the generated zip files onto the target machine(s).

4. **Start the framework and services**.

### Finding microservice logs

If a microservice framework node is running on a Windows host, by default logs of microservice instances on the node are written to the location *%USERPROFILE%\Siemens\logs\TcMSF*. If the environment variable **SIEMENS_LOGGING_ROOT** is defined, then the logs are written to the location *%SIEMENS_LOGGING_ROOT%\TcMSF*.

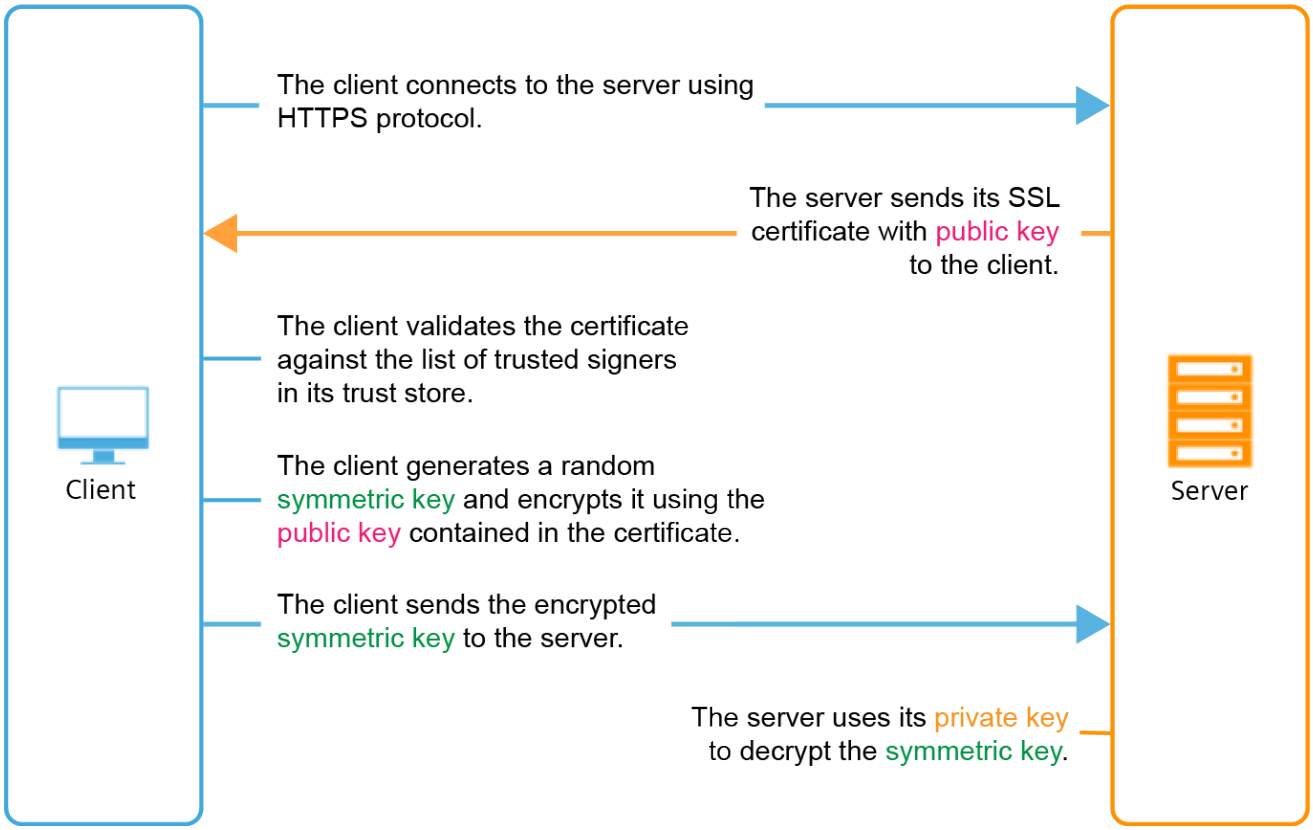
| Service | Log file name | Example |
|---|---|---|
| Service Dispatcher | service_dispatcher*instance#@PID*-msf.log | service_dispatcher1@2184-msf.log |
| Service Registry | eureka*instance#@PID*-msf.log | eureka1@2184-msf.log |
| Microservices | *microservice-name instance#@PID*-msf.log | file-repo1@2184-msf.log |
| Teamcenter Process Manager | *TC_ROOT*\process_manager\mgr.output | |
| web client gateway | gateway*instance#@PID*-msf.log | gateway1@2184-msf.log |

## Securing microservices

### Encrypting microservices traffic

An administrator can configure the microservice framework to encrypt data traffic based on an SSL certificate.

> Note:
>
> For a Kubernetes container manager on Linux, encryption can be configured for the ingress controller and service mesh. Refer to the documentation for the specific ingress controller or service mesh. Example service mesh implementations are Istio and Linkerd.




The server and client encrypt all further communication sent in either direction using the symmetric key. At the end of the session, the symmetric key is discarded.

Configuring the microservice framework and microservices for encrypted communication requires the following:

- Obtain an SSL certificate and keys for the server that will host the service dispatcher.

  A server certificate signed by a certificate authority (CA) can be purchased from a CA, and is recommended. Alternatively, cryptographic tools such as OpenSSL can be used to create a self-signed certificate and its keys. In the case of a self-signed certificate, the certificate issuer must be added to the client machine's trust store.

- When configuring a microservice node, for the **Service Dispatcher Setting**, choose the **HTTPS** protocol and complete the configuration parameters in the **HTTPS Config** component.

- When configuring the web client gateway, if you choose to override the default service dispatcher URL, ensure that you enter the HTTPS protocol for the **Service Dispatcher URL**.

- When deploying the container registry on Linux, for example, Docker Registry, **ensure that the container registry uses the HTTPS encryption protocol**.

### Configure microservices for self-signed certificates

If the service dispatcher is configured for HTTPS and the TLS certificate used is self-signed, each microservice on a node must be configured to trust the self-signed certificate. The following instructions apply to microservices that communicate with a server using a self-signed certificate. These steps describe how to configure the microservices with the Certificate Authority.

**Table 3-6. Configuring microservices on Windows systems**

| For these microservices | Do this |
|---|---|
| iModel Viewer Service<br>*(Source code language: Javascript/ Typescript using NodeJS)* | Prerequisite: The certificate must be in the PEM format and must not have been generated using DSA encryption.<br>1. Edit the file **%TC_ROOT%\microservices\services_config\<microservice>.json**.<br>2. In the **environment** section, add the environment variable **NODE_EXTRA_CA_CERTS** and set it to point to the location of the certificate.<br>Example:<br>  <br>darsi.json<br>{<br>"darsi": {<br>"image": "darsi-1.3.0",<br>"environment": [<br>"DSP=https://vc6s004:9090",<br>"MSR=https://vc6s004:8787/eureka/v2",<br>"NODE_EXTRA_CA_CERTS=C:/apps/tc/tc13/mytruststore.pem"<br>],<br>"deploy": {<br>"replicas": 1<br>}<br>}<br>}<br>For additional information about this variable, see NodeJS documentation.<br>3. Restart the process manager. |
| ep-app<br>FileRepo<br>mfe-vis<br>odata_service | 1. Ensure that the following two arguments are passed to the JVM:<br>`-Djavax.net.ssl.trustStorePassword=`*password*<br>`-Djavax.net.ssl.trustStore=`*path_to_trust_store_file_in_.jks_format*<br>The method for doing this for Java-based microservices depends on their implementation. |

| For these microservices | Do this |
| --- | --- |
| req-compare-service<br><br>*(Source code language: Java)* | • The preferred method is to edit the *TC_ROOT***microservices\services_config*****microservice****.json** file to alter the JVM arguments.<br><br>For most microservices, the **.json** file has an ARGS variable, to which you can append arguments.<br><br>• Some microservices, notably odata_service, require that you modify the corresponding *TC_ROOT***microservices*****microservice***start_service.bat** script to add the JVM arguments.<br><br>2. Restart the process manager. |
| Command Prediction<br><br>Google Online<br><br>Office Online<br><br>Product Configurator Service<br><br>reqexportservice<br><br>reqimportservice<br><br>Teamcenter Share<br><br>*(Source code language: C#)* | 1. If the trust store file is in **.jks** format, convert the **.jks** file to **.pk12**.<br><br>To convert a keystore file named **keystore2.jks** to a **.pk12** file using the key **mykey** and the password **testKeyStorepw**, run the command:<br><br>`keytool -importkeystore -srckeystore [./keystore2.jks] -destkeystore ./keystore2.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass testKeyStorepw -deststorepass testKeyStorepw -srcalias mykey -destalias mykey -srckeypass testKeyStorepw -destkeypass testKeyStorepw -noprompt`<br><br>2. Double-click the .pk12 file to install it as a trusted certificate. |

**Table 3-7. Configuring microservices on Linux systems**

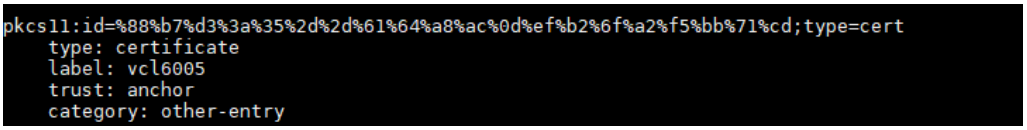
| For these microservices | Do this |
| --- | --- |
| iModel Viewer Service | Prerequisite: The certificate must be in the PEM format and must not have been generated using DSA encryption.<br><br>1. Edit the microservice configuration (YAML) file.<br><br>2. In the **environment** section, add the environment variable **NODE_EXTRA_CA_CERTS** and set it to point to the location of the certificate. |

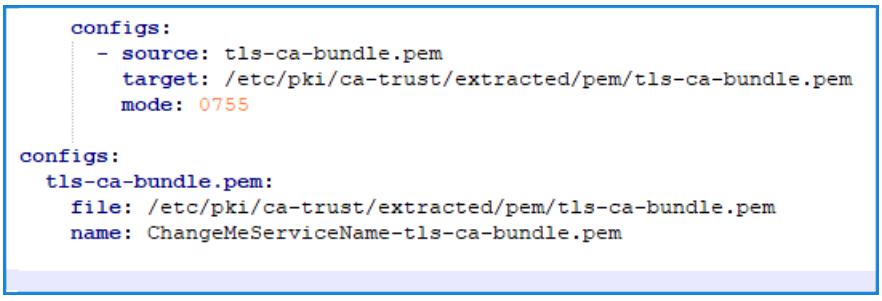
<table>
<tr><th>For these microservices</th><th>Do this</th></tr>
<tr><td>(Source code language: Javascript/ Typescript using NodeJS)</td><td>
Example:
<pre>version: "3.3"
services:
  darsi:
    image: myCorp:5000/teamcenter/afx-darsi:1.6.5
    deploy:
      mode: replicated
      replicas: 1
    environment:
      - FSC_URL=http://service_dispatcher:9090/
filerepo
      - MSR=http://eureka:8080/eureka/v2/
      - NODE_ENV=production
      - NODE_EXTRA_CA_CERTS=/run/secrets/cert_file_name.pem
    logging:
      driver: fluentd
      options:
        fluentd-address: 0.0.0.0:24223
        fluentd-async-connect: 'true'
        tag: 'msf.{{.Name}}.{{.ID}}'
    depends_on:
      - eureka
    secrets:
      - validator_keystore.pem
      - cert_file_name.pem

secrets:
  validator_keystore.pem:
    file: ./secrets/validator_keystore.pem
  cert_file_name.pem:
    file: ./secrets/cert_file_name.pem</pre>
For additional information about this variable, see NodeJS documentation.<br><br>
3. To update a running container image, deploy the updated ChangeMeServiceName microservice files.
</td></tr>
<tr><td>ep-app<br><br>FileRepo<br><br>mfe-vis</td><td>1. Ensure that the following arguments are passed to the JVM:<br><br><pre>-Djavax.net.ssl.trustStorePassword=trust_store_password</pre></td></tr>
</table>

| For these microservices | Do this |
| --- | --- |
| odata_service<br><br>req-compare-service<br><br>*(Source code language: Java)* | `-Djavax.net.ssl.trustStoreType=`*trust_store_type*<br><br>Note:<br>Enter the appropriate type, one of jks or pkcs12.<br><br>`-Djavax.net.ssl.trustStore=`*path_to_truststore_file*<br><br>Example:<br>`version: "3.3"`<br>`services:`<br>`  filerepo:`<br>`    hostname: filerepo`<br>`    image: vcl6005:5000/teamcenter/file-repo:6.3.0`<br>`    user: 0:0`<br>`    deploy:`<br>`      mode: replicated`<br>`      replicas: 1`<br>`    volumes:`<br>`      - /scratch/msf/filerepo:/fms/fsc/volume`<br>`    #@#logging:`<br>`    #@#  driver: fluentd`<br>`    #@#  options:`<br>`    #@#    fluentd-address: 0.0.0.0:24223`<br>`    #@#    fluentd-async-connect: 'true'`<br>`    #@#    tag: 'javamld.{{.Name}}.{{.ID}}'`<br>`    environment:`<br>`      - ARGS=-Deureka.serviceUrl.default=http://eureka:8080/eureka/v2/`<br>`          -Dsecrets_path=../../run/secrets/`<br>`          -DdispatcherUrls=https://vcl6005.net.plm.eds.com:9090/`<br>`          -Djavax.net.ssl.trustStorePassword=private`<br>`    -Djavax.net.ssl.trustStore=/run/secrets/`*trust_store_file_name*`.p12`<br>`    -Djavax.net.ssl.trustStoreType=pkcs12`<br>`    secrets:`<br>`      - tc_micro_security.properties`<br>`      - validator_keystore.p12`<br>`      - signer_tc_micro_security.properties`<br>`      - signer_keystore.p12`<br>`  - `*trust_store_file_name*`.p12`<br>`    depends_on:`<br>`      - eureka` |

<table>
<tr><th>For these microservices</th><th>Do this</th></tr>
<tr><td></td><td>
<pre>secrets:
  tc_micro_security.properties:
    file: ./secrets/tc_micro_security.properties
  validator_keystore.p12:
    file: ./secrets/validator_keystore.p12
  signer_tc_micro_security.properties:
    file: ./secrets/
signer_tc_micro_security.properties
  signer_keystore.p12:
    file: ./secrets/signer_keystore.p12
  trust_store_file_name.p12:
  file: ./secrets/trust_store_file_name.p12</pre>
2. To update a running container image, deploy the updated ChangeMeServiceName microservice configuration (YAML) file.
</td></tr>
<tr><td>Google Online<br>Office Online</td><td>
1. Copy your self-signed certificate in PEM format to the appropriate location depending on the host operating system.
<table>
<tr><th>For this operating system</th><th>Use this location</th></tr>
<tr><td>Red Hat</td><td>/etc/pki/ca-trust/source/anchors</td></tr>
<tr><td>Suse</td><td>/usr/share/pki/trust/anchors/</td></tr>
</table>
2. Install the certificate on the host operating system.
<table>
<tr><th>For this operating system</th><th>Run this command</th></tr>
<tr><td>Red Hat</td><td><code>update-ca-trust</code></td></tr>
<tr><td>Suse</td><td><code>sudo update-ca-certificates</code></td></tr>
</table>
3. To verify that the certificate is installed, run the command trust list and check that your certificate is in the list.
<pre>pkcs11:id=%88%b7%d3%3a%35%2d%2d%61%64%a8%ac%0d%ef%b2%6f%a2%f5%bb%71%cd;type=cert
    type: certificate
    label: vcl6005
    trust: anchor
    category: other-entry</pre>
4. Run the trust utility to generate a ca bundle file.
<pre>sudo trust extract --filter=certificates
--format=pem-bundle /location/to/tcroot/microservices/
container/secrets/tls-ca-bundle.pem</pre>
</td></tr>
<tr><td>Command Prediction</td><td>1. Copy your self-signed certificate in PEM format to the appropriate location depending on the host operating system.</td></tr>
</table>

<table>
<tr><th>For these microservices</th><th>Do this</th></tr>
<tr><td>Product Configurator Service<br><br>reqexportservice<br><br>reqimportservice<br><br>Teamcenter Share<br><br>(Source code language: C#)</td><td>
<table>
<tr><th>For this operating system</th><th>Use this location</th></tr>
<tr><td>Red Hat</td><td>/etc/pki/ca-trust/source/anchors</td></tr>
<tr><td>Suse</td><td>/usr/share/pki/trust/anchors/</td></tr>
</table>
2. Install the certificate on the host operating system.
<table>
<tr><th>For this operating system</th><th>Run this command</th></tr>
<tr><td>Red Hat</td><td><code>update-ca-trust</code></td></tr>
<tr><td>Suse</td><td><code>sudo update-ca-certificates</code></td></tr>
</table>
3. To verify that the certificate is installed, run the command trust list and check that your certificate is in the list.


4. Run the trust utility to generate a ca bundle file.
<pre>sudo trust extract --filter=certificates
--format=pem-bundle /location/to/tcroot/microservices/
container/secrets/tls-ca-bundle.pem</pre>
5. Edit each microservice configuration file (YAML) to include a config object for the updated CA certs file. Replace occurrences of ChangeMeServiceName with the microservice name.


6. To update a running container image, deploy the updated ChangeMeServiceName microservice files.
</td></tr>
</table>

## High availability for microservices

In a distributed Teamcenter production environment, ensure high availability by configuring redundant microservice node servers and service instances. For detailed deployment examples and sample configurations, see *Teamcenter Deployment Reference Architecture*, available from the Teamcenter documentation and also from the **Support White Papers Teamcenter Deployment Reference Architecture** page on Support Center.

### Capacity

With the many variables affecting a Teamcenter environment, no simple formula exists that can prescribe the precise combination of microservice nodes and microservice instances. As with all server-side deployments, monitor the consumption of CPU and memory on each microservice node. If you observe resource contention, you can increase resources for microservice execution by deploying additional microservice nodes and services running on additional hardware.

### Failover

#### Windows

Achieving failover capability on Windows requires that a service registry, a service dispatcher, and instances of all microservices must each be running on at least two nodes. By default, an instance of the service registry and service dispatcher run on the master node; additional instances can be running on any worker nodes. When installing microservice nodes through Deployment Center, be sure to list all instances of the service registry and the service dispatcher.

#### Docker Swarm

Achieving failover capability with Docker Swarm on Linux requires that an odd number of servers be joined to the swarm as managers, typically three or five. This helps the Docker swarm effectively manage the swarm by majority vote. Any number of servers can be joined to the swarm as workers.

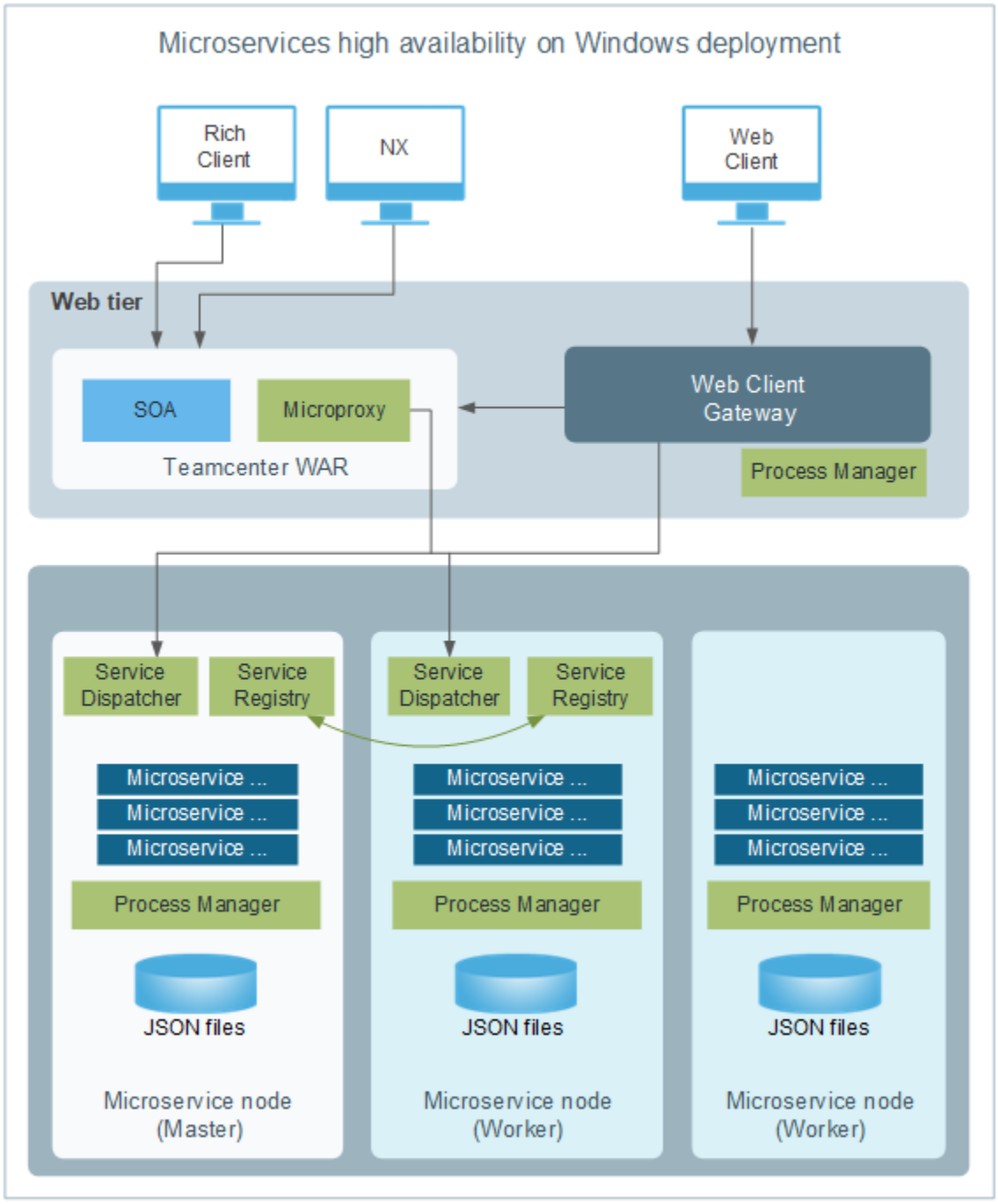
#### Kubernetes

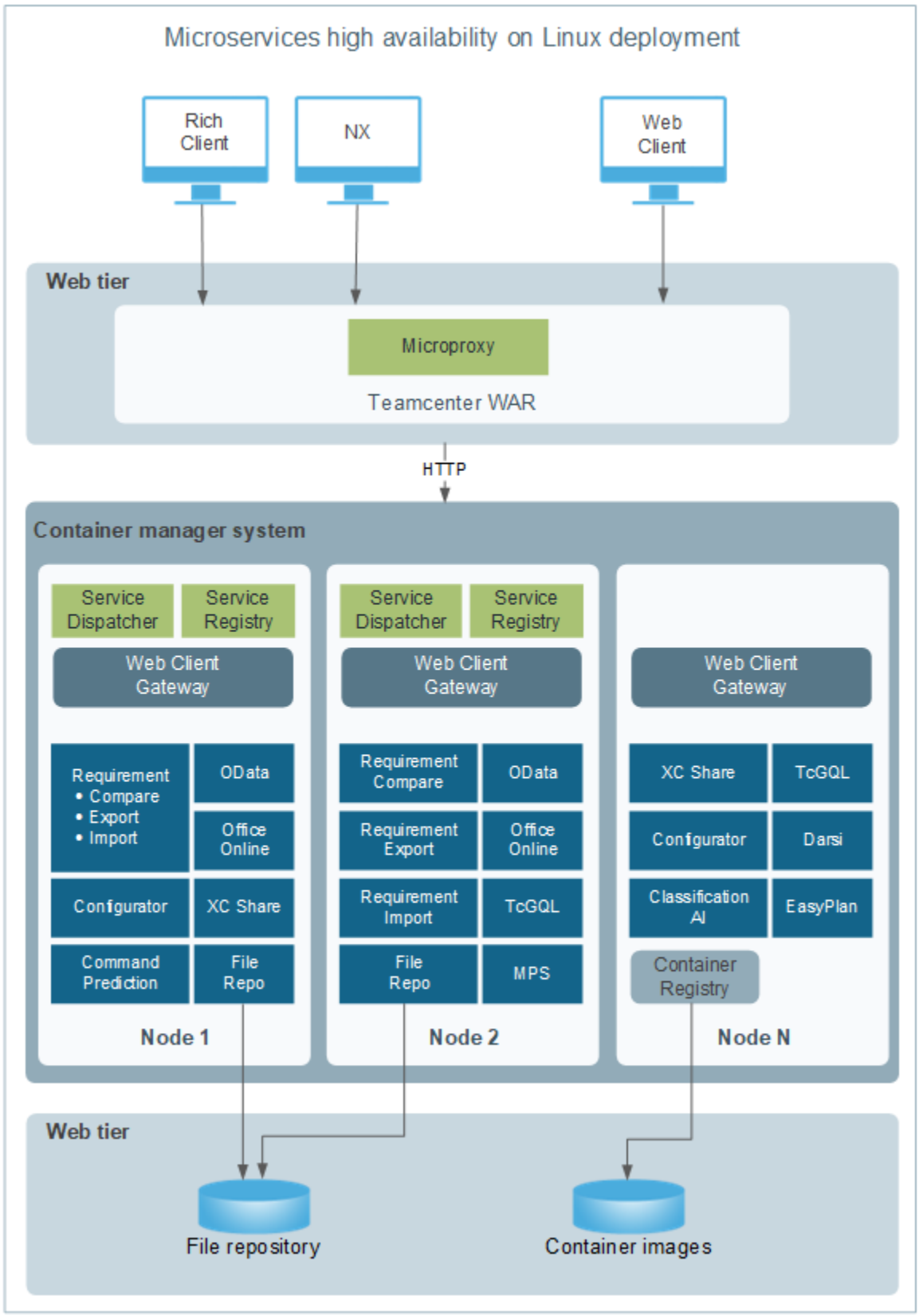
| | |
|---|---|
| **Control Plane** | Follow the vendor documentation. If using a cloud provider, the provider typically provides a Control Plane with failover. |
| **Microservice nodes** | To avoid a single point of failure, in on-site deployments implement at least two microservice nodes. Ensure that these nodes are allocated on different physical hardware. Allocate at least two replicas of every component to avoid a single point of failure. For nodes in cloud deployments, to avoid location-specific outages, ensure that the nodes are spread across different failure zones (such as AWS Availability Zones).<br><br>The exception to replicating components is the Service Registry. A single Service Registry is sufficient. This is because in the event that the Service Registry (Eureka) container goes down, the Eureka Client Cache provides needed information during the brief period of time that passes while the container manager brings back up the container. |

If possible, test for node failure conditions and validate that client requests are handled using service load balancing. Ensure desired scale once the nodes are recovered.

For backup options, consult the vendor documentation.

### Example microservice deployment topologies for high availability





In a high availability configuration, where there is more than one File Repository microservice deployed, with the File Repository storage shared between the multiple instances of File Repository, the Active Workspace publish operation needs to be performed only once, from one of the nodes (preferably the primary node). This updates the shared storage of the File Repository microservice, so the publish does not need to be repeated from another node. Any new publish will overwrite the previous publish in the shared storage. Additionally, the publish should always be done from the same node (for example, the primary node) as used for the previous publish. It is not recommended to publish from a different node in a subsequent publish, as this may cause errors.

## Install microservices

The **microservice framework** must be installed before you begin these steps. You can add microservices to an existing microservice node or install the microservices and Microservice Framework at the same time.

1. Log on to Deployment Center and select your Teamcenter environment.

2. In the **Software** tab, make sure the **Selected Software** list includes the **Teamcenter 2412** software.

3. In the **Options** tab, choose the **Distributed** environment type.

4. Proceed to the **Components** tab. Select the **Microservice Node** component and enter required values for the following microservices:

**Table 3-8. File Repository Service**

| Value | Description |
|---|---|
| **File Repository Storage Location** | Type a location for the file repository to be used by the web client gateway. The path must exist on the machine that hosts the microservice node. For example: **c:\tc\file_repository**<br><br>The file repository stores web client content. |
| **User ID** | (Linux only) Type the user ID of the user installing the File Repository Microservice. |
| **Group ID** | (Linux only) Type the Group ID of the user installing the File Repository Microservice. |

**Table 3-9. Teamcenter GraphQL Service**

| Value | Description |
|---|---|
| **Teamcenter Web Tier URL** | In Deployment Center, the **Microservice Node** locates the URL from the **Teamcenter Web Tier** component automatically. |

5. Continue with the **web client gateway installation**.

Note:

On Linux systems, microservices Worker Nodes must contain the same installed microservices as the Master Node.

## Installing Security Services

Teamcenter Security Services (TcSS) provides integrated login, authentication and single sign-on (SSO) services for Teamcenter and its application suite.

The Security Services Login Service and Identity Service are Java EE web applications (WAR files) that provide the essential functions of Security Services. Deployment Center builds these applications, which you then deploy on a supported Java EE web application server.

For information about supported application servers and Java versions, see the Hardware and Software Certifications knowledge base article on Support Center.

This procedure assumes you have an existing Teamcenter environment. Make sure the required Teamcenter software kits have been added to your software repository in Deployment Center.

Log on to Deployment Center and select your Teamcenter environment, then begin installing TcSS.

### Configure TcSS LDAP servers

If you build the TcSS Login and Identity Services using Deployment Center and you have an LDAP server set up and want to configure LDAP settings in the Identity Service, you must configure the **TcSS LDAP** component.

If you build the TcSS Login and Identity Services using the Web Application Manager (**insweb**), skip these steps and proceed to **Configure Teamcenter Security Services**.

1. In the **Components** tab in Deployment Center, click **Add component to your environment** ⊕ to display the **Available Components** panel.

2. Select **TcSS LDAP**, and then click **Update Selected Components**.

3. In the **Selected Components** list, select **TcSS LDAP**.

4. Enter **Machine Name** and **OS** values for the machine on which you install Security Services:

   - **Single box**

     If your environment is a **single box** environment, the **Machine Name** and **OS** values are inherited from the first component you configured in your environment. Changing these values will change them for other components in your environment.

- **Distributed**

  If your environment is a **distributed** environment, select a machine name from the dropdown list or enter a new machine name. Then, enter the **OS** for the machine on which you install Security Services.

5. Under **External LDAP Settings**, enter required parameters:

| Parameter | Description |
|---|---|
| **LDAP Ordinal** | Specifies the search order for the LDAP server. The server with the lowest number (for example, **1**) indicates the *primary LDAP server*.<br>Other numbers indicate the order in which secondary LDAP servers are searched, from lowest to highest. |
| **LDAP Protocol** | Specifies the protocol to use when connecting to LDAP servers.<br>Selecting **tls** specifies non-SSL LDAP connections, but then uses **startTLS** protocol to promote the connection to TLS.<br>Selecting **ldap** specifies non-SSL LDAP connections.<br>Selecting **ldaps** specifies SSL LDAP connections.<br>Selecting **auto** allows the system to determine the connection type dynamically for each LDAP server. |
| **Port** | Specifies the port number to use to connect to the LDAP server. |
| **Administrator DN** | Specifies the distinguished name (DN) used to authenticate to the LDAP server for LDAP searches. |
| **Administrator Password** | Specifies the password for the Administrator DN. Retype the password in the **Confirm Administrator Password** box.<br>The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab.<br>In addition, the password must not contain any of the following characters:<br>**! # @ $ % = & ' " ^ : ; . _ < > ( ) { }** |
| **Max LDAP Connections** | Specifies the maximum number of connections that can be created per Identity Service instance for each LDAP server.<br>Larger values mean lower resource contention at the expense of higher resource consumption. Smaller values conserve resources but may cause some blocking during login due to resource contention. In a clustered environment with many Identity Service instances, a value between **2** and **20** is recommended. |

| Parameter | Description |
| --- | --- |
| **LDAP Connection Setup Delay** | Specifies the interval in seconds to wait between initiating a parallel connection to each successive server in the list. This applies when multiple LDAP servers are specified or when multiple Domain Controllers are discovered via DNS lookup.<br>A value of **-1** means connect to the servers serially, and **0** means initiate parallel connections to all servers at once. |
| **LDAP Connection Timeout** | Specifies the interval in seconds to wait before abandoning an LDAP request, which can include connection, search, and bind attempts. If **LDAP Connection Setup Delay** is greater than **0**, this value should be greater in order to allow multiple connection attempts.<br>A value of **0** means unlimited. |

> Note:
>
> Do not include the pound sign (**#**) or semicolon (**;**) in *any* parameters in the **TcSS LDAP** component.

For a complete mapping of Security Services context parameters from the Web Application Manager (**insweb**) to Security Services properties in Deployment Center, see **Security Services properties in Deployment Center**.

6. Click **Save Component Settings**.

For more information about configuring LDAP servers, see *Security Services Configuration*.

### Configure Teamcenter Security Services

1. In the **Components** tab in Deployment Center, click **Add component to your environment** ⊕ to display the **Available Components** panel.

2. Select **Teamcenter Security Services (TcSS)**, and then click **Update Selected Components**.

> Note:
>
> Do not include the pound sign (**#**) or semicolon (**;**) in *any* parameters in the **Teamcenter Security Services (TcSS)** component.

3. In the **Selected Components** list, select **Teamcenter Security Services (TcSS)**.

4. Specify the **Machine Name** and **OS** values as appropriate for your environment type.

5. Under **TcSS Settings**, enter credentials for the **Teamcenter Administrative User in TcSS LDAP**:

| | |
|---|---|
| **User** | Specifies the administrative user configured in your LDAP server. |
| **Password** | Specifies the password for the LDAP administrative user. Retype the password in the **Confirm password** box. |

> Tip:
>
> To locate a specific parameter, use your web browser's search function.

6. Select the **Use Deployment Center to build the Login Service and Identity Service WAR files** check box.

> Note:
>
> Alternatively, you can build the Security Services login service and identity service using the Web Application Manager as described in the Teamcenter installation guides for TEM (for Windows or Linux). This will require additional parameter settings in Deployment Center to use the Security Services WAR files generated by the Web Application Manager.

7. In the **Staging Location** parameter, type the path in which to place the generated WAR files, for example, **c:\staging**.

   When you generate deployment scripts, Deployment Center places TcSS WAR files in a **deployment** subdirectory beneath the staging location you specify.

8. Enter remaining required values to configure Security Services:

| Value | Description |
|---|---|
| **Common WAR File Settings** | Settings to configure the properties password for Security Services. |
| **Properties Password** | Specifies the password that will be used to decrypt encrypted property values stored in properties files (for example, **federation.properties**). Retype the password in the **Confirm Properties Password** box.<br>The password must not be empty nor contain any whitespace characters such as space, tab, newline, carriage return, form feed, or vertical tab.<br>In addition, the password must not contain any of the following characters:<br>**! # @ $ % = & ' " ^ : ; . _ < > ( ) { }** |
| **TcSS Login URL Settings** | Settings to configure the Security Services Login Service. |

| Value | | Description |
|---|---|---|
| | **Login Service's Web App Server Machine Name** | Specifies the name of the machine on which you deploy the Security Services Login Service WAR file. |
| | **Protocol** | Specifies the protocol to use to connect to the web tier (**http** or **https**).<br>If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| | **Port** | Specifies the port through which the Login Service connects. The default value is **7001**. |
| | **Login Service Application Name** | Specifies the name of the Login Service, for example, **tcssols**. |
| | **Login Service URL** | Specifies the URL to the Login Service. This parameter is not directly editable, but is constructed from the protocol, port, machine name, and application name of the Login Service, for example, **http://myHost:7001/tcssols** |
| | **Identity Service Password** | Type a password for connecting to the TcSS Identity Service. Retype the password in the **Confirm Identity Server Password** box. |
| **TcSS Identity Service URL Settings** | | Settings to configure the Security Services Identity Service. |
| | **Login Service's Web App Server Machine Name** | Specifies the name of the machine on which you deploy the Security Services Identity Service WAR file. |
| | **Protocol** | Specifies the protocol to use to connect to the web tier (**http** or **https**).<br>If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| | **Port** | Specifies the port through which the Identity Service connects. The default value is **7001**. |
| | **Login Service Application Name** | Specifies the name of the Login Service, for example, **tcssoservice**. |
| | **Login Service URL** | Specifies the URL to the Login Service. This parameter is not directly editable, but is constructed from the protocol, port, machine name, and application name of the Login Service, for example, **http://myHost:7001/tcssoservice** |
| | **Mediator Password** | Specifies a password shared between the Identity Service and a mediating application. Retype the password in the **Confirm Mediator Password** box.<br>This password is used to encrypt tokens passed to the mediator for later distribution to applications participating in trust relationships. |

For a complete mapping of Security Services context parameters from the Web Application Manager (**insweb**) to Security Services properties in Deployment Center, see **Security Services properties in Deployment Center**.

If you want to specify additional settings for Security Services, click **Show all parameters** .

When you are finished configuring TcSS parameters, click **Save Component Settings**.

9. Complete configuration of any remaining components.

10. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

11. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see the *Deployment Center — Usage*.

12. In the **deployment** directory under your specified staging location on the target machine, find the Login Service and Identity Service WAR files.

13. Deploy the web applications on a supported application server.[1]

For more information about configuring Security Services to meet your security requirements, see *Security Services Configuration*.

To see how Security Services context parameters from the Web Application Manager (**insweb**) map to Security Services properties in Deployment Center, see **Security Services properties in Deployment Center**.

## Install Active Workspace Gateway

Active Workspace Gateway requires the keystore ZIP file (**keys.zip**) from the microservice master node. Before you install Active Workspace Gateway, copy the **keys.zip** file from the **jwt_config_tool** directory under *TC_ROOT* on the microservice master node host to a directory on the Active Workspace Gateway host.

You can install Active Workspace Gateway in a new or an existing Teamcenter environment.

> Note:
>
> Install Active Workspace Gateway using the same tool with which you **install Active Workspace microservices**. For example, if you install microservices using Deployment Center, install Gateway using Deployment Center. Or, use TEM for both installations.
>
> When entering URLs, ensure you use URLs with fully qualified domain names or IP addresses.

1 *Web Application Deployment* provides Teamcenter web tier deployment procedures for several supported application servers.

1. Make sure you have the Teamcenter 2412 software in your repository.

   Choose your new or existing environment, and create or update the **Selected Software** list.

2. In **Selected Applications**, required **Active Workspace** applications are automatically listed.

   Add the following applications to your environment:

   - **Teamcenter Share Collaboration**
   - **Teamcenter Share Collaboration Active Workspace**

3. In the **Selected Components** list, choose **Active Workspace Gateway**.

   Enter the machine name and operating system. The installation path to Teamcenter may be specified automatically if it was entered in another component.

4. Expand the configuration sections to show all parameters, and enter the required values.

| Value | Description |
|---|---|
| **Port** | Enter the port for Active Workspace Gateway. The default value is **3000**.<br>The URL to the Active Workspace client interface will use this port. |
| **Use SSL protocol** | Specifies the protocol to use to connect to the gateway (**http** or **https**).<br>If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| **Use as Bootstrap URLs** | The Active Workspace client uses FMS to download and upload files. You define the FSC servers that are used by selecting either **Use as Bootstrap URLS** or **Use Assigned FSC URLs**.<br>On Linux hosts, you must select **Use as Bootstrap URLs** to ensure the client map is configured correctly. |
| **Bootstrap Client IP** | Specifies the FMS bootstrap client IP address to be used for the assignment.<br>On Linux hosts, enter the internal IP address of the Active Workspace Gateway machine. |
| **Use Assigned FSC URLs** | Specifies whether you want to assign FSC servers. Select this only if you want explicit control of the FSCs used. |
| **FSC Connection URL** | Specifies a comma-separated list of URLs to one or more existing FMS server caches (FSCs).<br>The URL must be of the form:<br>**http://***host***:***port* |

| Value | Description |
|---|---|
| | By default, the IP address from the HTTP connection of the requestor is used unless a **Bootstrap Client IP** value is provided. (The client/requestor is the host on which Active Workspace Gateway is deployed.) |
| **Assigned FSC URLs** | Specifies a comma-separated list of one or more assigned FSC URL values.<br>The URL values entered are directly used for file operations. This allows you to declare the FSC servers that should be used. |

Service Dispatcher URLs are obtained from the Microservice Node. The Service Dispatcher URLs must include fully qualified domain names or IP addresses.

You may choose whether to communicate with Teamcenter through the Teamcenter web tier or through a load balancer. Specify your settings in the **Teamcenter Server Connection Settings** section.

5. Under **Teamcenter Share Collaboration**, type your Teamcenter Share Collaboration settings:

| Value | Description |
|---|---|
| **Teamcenter Share URL** | Specifies the URL to the Teamcenter Share site.<br>The default value is **https://share.sws.siemens.com**. |
| **Client ID** | Specifies the SAMAuth client ID you obtained through SAM URL. |
| **Client Secret** | Specifies the client secret ID you obtained through SAM URL. Enter this value again in the **Confirm Client Secret** box to confirm. |

6. You may specify the configuration for other components now or later. Proceed to **installing the Active Workspace client** for instructions.

> Note:
>
> Verify the FMS server cache (FSC) service is running before you start the Active Workspace Gateway service.

7. (If Active Workspace Gateway is deployed on a Linux host) Start the Docker swarm, microservices node, and the Gateway service.

## Install the Active Workspace client

Before you install the Active Workspace client using Deployment Center, you must complete the following:

- *Install microservices*

- *Install Active Workspace Gateway*

You can install the Active Workspace client concurrently with Active Workspace Gateway.

Install the Active Workspace client configuration using the Teamcenter 2412 software kit.

### Install the Active Workspace client

1. Selecting the Active Workspace software automatically includes its basic applications in the **Selected Applications** list. The associated components required to deploy Active Workspace are listed in the **Selected Components**.

   If you haven't already, you can select additional applications you want to include in your Active Workspace environment.

2. In the **Selected Components** list, choose **Active Workspace Client Builder**.

   Enter the machine name and operating system. The installation path to Teamcenter may be specified automatically if it was entered in another component.

3. If you want to automatically publish Active Workspace content to the Gateway, check **Publish Active Workspace Client Assets**.

   > Note:
   >
   > The Active Workspace Gateway must be installed *and* running before content can be published.

4. When the remaining component configuration is complete, click **Go to Deploy** and generate your deployment scripts.

## Installing indexing components

The Indexing Engine and the Indexer provide global search capabilities for Active Workspace. Install these components using Deployment Center:

- **Indexing Engine**

  Installs the Solr enterprise search platform. The search engine stores indexed Teamcenter data for global search in Active Workspace.

  Selected product data is indexed in Solr, an open source search platform from Apache. The master product data is not stored in Solr. It is always loaded from Teamcenter.

- **Indexer**

Installs a four-tier SOA client that exports Teamcenter data for merging into Solr. The indexer manages overall indexing processes. **TcFTSIndexer** manages the initial indexing for object data. You can then schedule synchronization to run periodically for subsequent updates to object data or structure data indexes.

TcFTSIndexer indexes external and Teamcenter objects into Solr. It connects to the server manager to query and extract Teamcenter data to be indexed into Solr.

There are two modes for installing the **Indexer**: **Standalone** for object data and **Dispatcher-based** for Active content structures.

- **Asynchronous File Content Indexing**

  Installs an optional feature that allows you to index file contents asynchronously from object metadata. Dispatcher is used to manage file content requests.

- **Teamcenter Artificial Intelligence (AI) Chat**

  Installs an optional feature that allows you to ask natural language questions and receive summarized answers with source material.

For information about indexing components and planning your indexing deployment, see the *Indexing Data and Configuring Search* in the Active Workspace documentation.

During installation of indexing components, you may need to enter or verify the following information:

| **Indexing Engine Configuration** | | |
|---|---|---|
| **Parameter** | **Where value is defined** | **Your value** |
| Teamcenter machine name and installation path | Teamcenter installation | |
| *TC_DATA* | Teamcenter installation | |
| Java location | Indexing Engine installation | |
| Solr directory and credentials | Indexing Engine installation | |
| Solr URL (http://host:8983/solr) | Indexing Engine installation | |
| Indexing Engine credentials | Indexing Engine installation | |

## Install Indexing Engine (Solr)

You can install the Indexing Engine (Solr) in a new or existing environment.

### Prerequisites

This procedure assumes you have an existing Teamcenter environment with Active Workspace.

Make sure all the required software kits have been added to your software repository in Deployment Center

### Procedure

1. Log on to Deployment Center and select your Teamcenter environment.
2. In the **Software** tab, make sure the **Selected Software** list includes Teamcenter 2412.
3. If you want to install the Indexing Engine with SolrCloud, in the **Options** tab:
   - Select **Distributed** in **Environment Type**.
   - Select the **High Availability** check box, which installs a high availability SolrCloud configuration.
4. Proceed to the **Components** tab.

   If the **Selected Components** list does not include **Indexing Engine**, add this component:

   a. Click **Add component to your environment** to display the **Available Components** panel.

   b. Select **Indexing Engine**, and then click **Update Selected Components**.
5. In the **Selected Components** list, select **Indexing Engine**.

   If you selected a distributed environment with a high availability SolrCloud configuration, you must configure both the **Indexing Engine** and **Indexing Engine (HA)** components for different machines.
6. In the **Indexing Engine** panel(s), enter values for the following configuration parameters:

| Parameter | Description |
|---|---|
| **Machine Name** | Specifies the name of the machine on which the Indexing Engine is installed. |
| **OS** | Specifies the operating system on which the Indexing Engine is installed. |

| Parameter | Description |
|---|---|
| **Teamcenter Installation Path** | Specifies the installation path for the machine on which in the Indexing Engine is installed. |
| **Indexing Engine User** | Type the user name and password for the Solr administrator. These credentials must match the Indexer and the Active Content Structure Translator (if used). |
| **Install Indexing Engine as a Service** | Select the Install Indexing Engine as a Service check box if you want to install the Indexing Engine as a service.<br>If you clear this check box, you must start the Indexing Engine manually after deployment on the Indexing Engine machine. |
| **Operating System User** | Type the operating system user name and password on the Indexing Engine machine.<br>If the Indexing Engine machine is a Windows machine, include the domain name (**domain\user**). |

If you want to specify additional settings for the Indexing Engine, click **Show all parameters**. To configure Solr for HTTPS, select **https** in **Server Protocol** and complete the configuration parameters in the **HTTPS Config** component.

7. If you selected a distributed environment with a high availability SolrCloud configuration, you must configure all three **Zookeeper (HA)** components. In each **Zookeeper (HA)** panel, enter values for the following configuration parameters:

| Parameter | Description |
|---|---|
| **Machine Name** | Specifies the name of the machine on which Zookeeper is installed. |
| **OS** | Specifies the operating system on which Zookeeper is installed. |
| **Teamcenter Installation Path** | Specifies the installation path for the machine on which Zookeeper is installed. |
| **Zookeeper ID** | Specifies the unique identifier assigned to the Zookeeper component. |
| **Zookeeper Port** | Specifies the location where the Zookeeper process runs. |

8. When you finish entering values for all components, click **Save Component Settings**.

9. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** tab is enabled.

10. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

11. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

12. Migrate to SolrCloud using the SolrCloud utility.

13. Configure indexing and search after completing the Indexing Engine installation.

## Install the Indexer (TcFTSIndexer)

You can install the Indexer (TcFTSIndexer) in a new or existing environment.

### Prerequisites

This procedure assumes you have an existing Teamcenter environment with Active Workspace.

Make sure all the required software kits have been added to your software repository in Deployment Center.

### Procedure

1. Log on to Deployment Center and select your Teamcenter environment.

2. In the **Software** tab, make sure the **Selected Software** list includes Teamcenter 2412.

3. Proceed to the **Components** tab.

   If the **Selected Components** list does not include **Indexer**, add this component:

   a. Click **Add component to your environment** ⊕ to display the **Available Components** panel.

   b. Select **Indexer**, and then click **Update Selected Components**.

4. In the **Selected Components** list, select **Indexer**.

5. In the **Indexer** panel, enter values for the following configuration parameters:

| Parameter | Description |
|---|---|
| **Machine Name** | Specifies the name of the machine on which the Indexer is installed. |
| **OS** | Specifies the operating system on which the Indexer is installed. |

| Parameter | Description |
|---|---|
| **Teamcenter Installation Path** | Specifies the installation path for the machine on which in the Indexer is installed. |
| **Install Database Triggers for Indexing** | Select the **Install Database Triggers for Indexing** check box if you want to install database triggers. |
| **Maximum Teamcenter Connections** | Specifies the maximum number of connection between the Teamcenter server and the indexer that can be open at a given time.<br>This number should not exceed the number of warm TcServers available in Teamcenter server manager pool, and controls the performance of the indexing process using parallel steps. The default value is **3**. The minimum value is **2**.<br>Initially, consider the number of warm servers available in this environment and the percentage of them that are available for indexing only. |
| **Install Indexer as a Service** | Select the **Install Indexer as a Service** check box if you want to install the objdata synchronization flow and the suggestion builder synchronization flow of the indexer as services.<br>The **Service Name** fields populate with suggested names for the services, and can be edited.<br>The **Sync Interval** fields populate with suggested intervals for the synchronization flows and can be edited.<br>Select the **Start Service** check box to automatically start the service. |
| **Operating System User** | Type the operating system user name and password on the Indexer machine.<br>If the Indexer machine is a Windows machine, include the domain name (**domain\user**). |
| **Indexer Administrative User** | Select the **Set Indexer Administrative User Info** check box if you want to specify an administrative indexer user.<br>Type the administrator user name and password on the Indexer machine. |

If you want to specify additional settings for the Indexing Engine, click **Show all parameters**.

6. When you finish entering values for the Indexer component, click **Save Component Settings**.

7. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** tab is enabled.

8. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

9. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

10. Configure indexing and search after completing the Indexer installation.

## Install shape search

You can install the shape search feature on a new or existing environment to search for objects of a similar shape or size.

### Prerequisites

- This procedure assumes you have an existing Teamcenter environment with Active Workspace.
- Install, configure, and complete indexing for Geolus. See the **Geolus** documentation.
- Verify system software requirements:

  1. Log on to Support Center and open the **Support White Papers Certifications** page:

     a. Open **Products→Teamcenter→Downloads**.

     b. Under **Select a Version**, choose **Support White Papers→Support White Papers Certifications**, and then click the **Support White Papers Certifications** tile.

  2. Download the following support documents:

     **Software Certifications Matrix (Tc2412PlatformMatrix-*date*.xlsx)**

     Contains information about system software certified for Teamcenter, such as operating systems and Java runtime environments (JREs).

     **Teamcenter Interoperability Matrix (Teamcenter Interoperability Matrix *date*.xlsx).**

     Lists versions of Siemens Digital Industries Software products that are compatible with Teamcenter 2412. It also lists supported Teamcenter upgrade paths.

     Teamcenter 2412 supports upgrades from Teamcenter 13.*x* or later. If your current Teamcenter environment is using an earlier version than 13.*x*, you must upgrade to version 13.*x* or later before you upgrade to Teamcenter 2412.

     The Teamcenter Interoperability Matrix also correlates versions of Deployment Center with compatible versions of Teamcenter, and shows supported paths for upgrading Deployment

Center. For information about upgrading Deployment Center, see *Deployment Center — Usage*.

### Procedure

1. Add the **Shape Search** application to your Teamcenter environment:

   In Deployment Center, in the **Applications** tab, select **Shape Search**.

2. Enter configuration parameters in the **Components** tab.

   Enter the Geolus server URL in the following format: *protocol*://*gServer*:*gPort*/*gContext*

   - *protocol* can be **http** or **https**.
   - *gServer* is the machine name or IP address of the machine running the Geolus server. It must be accessible to all Teamcenter clients that need to connect to it.
   - *gPort* is the port number that the server uses to handle HTTP or HTTPS requests.
   - *gContext* is the context root of the Geolus server.

3. Generate and deploy the deployment scripts.

4. Configure shape search in NX so JT files are read in Teamcenter and shape search results are displayed in Active Workspace. For more information, see NX documentation.

   a. In NX, choose **File** > **Utilities** > **Customer Defaults** > **Gateway** > **JT Files** > **Export**, and select **Save JT Data**.

   b. In NX, choose **File** > **Preferences** > **Teamcenter Integration** > **Active Workspace**, and select **Display Shape Search Results in Active Workspace**.

## Install asynchronous file content indexing

The *asynchronous file content indexing* feature can be installed in a new or an existing Teamcenter environment using Deployment Center. Ensure that you have installed Teamcenter.

### Considerations

If you are installing the asynchronous file content indexing feature in an existing environment, consider the following:

- You must perform a full index of your existing data after installing asynchronous file content indexing for the first time.

- Additional storage may be required.

## Install asynchronous file content indexing

Perform the following steps to install asynchronous file content indexing through Deployment Center.

1. In the **Software** tab, ensure that Teamcenter 2412 is selected in the **Available Software** list.

2. In the **Applications** tab, click **Add or Remove Selected Applications** to add an application.

3. From the list of **Available Applications**, select **Asynchronous File Content Indexer** and click **Update Selected Applications**.

   In the list of **Selected Applications** under **Active Workspace**, **Asynchronous File Content Indexer** is now marked as **(Pending Install)**.

4. Click **Go to Components**.

   In the list of **Selected Components**, the following are now marked for install or update:

   - **Corporate Server**
   - **Dispatcher Client**
   - **Dispatcher Module**
   - **Indexer**
   - **Indexing Engine**

5. Select **Dispatcher Module**.

   In the **Translators** section, **Async File Content Indexing Translator** is selected automatically.

6. In the **Translators Settings** section, select options to enable the translators required for your system.

   The provided NX or Solid Edge extractors let you index the contents extracted from NX or Solid Edge CAD files. Enable and specify the location of the extractors you need for your system. For indexing standard files, such as Microsoft Office files, PDF files, and text files, no additional configuration is needed.

| Extractor | Location |
| --- | --- |
| NX | The extractor is shipped with NX. |

| Extractor | Location |
|---|---|
| | Extractor location: *NX-installation-path***NXBIN** |
| Solid Edge | Download the file from **Support Center > Solid Edge > Downloads > Solid Edge** *Year* **Add-ons > Teamcenter Feature Package > SEEC_Administrator_***Year***_MP5_Tc14.zip** and unzip to a temporary directory.<br>Extractor location: *temp_dir***SEEC_Administrator_***Year***_MP5_Tc14\Deep_CAD_Search_SE_Extractor** |

7. Click **Save Component Settings**.

8. In the **Selected Components** list, select **Indexer**.

9. Configure the Indexer component:

**Install Database Triggers for Indexing**
Choose whether to install database triggers for indexing.

**Maximum Teamcenter Connections**
This specifies the maximum number of connections between the Teamcenter server and the indexer that can be open at a given time.

This number should not exceed the number of warm TcServers available in Teamcenter server manager pool. This setting controls the performance of the indexing process using parallel steps. The default value is **3**. The minimum value is **2**.

> Tip:
>
> Initially, consider the number of warm servers available in this environment and the percentage of them that are available for indexing only.

If you want to specify additional settings for the Indexing Engine, click **Show all parameters** and enter these optional parameters:

**Teamcenter Retry Count**
This specifies the number of attempts the system allows to connect to the Teamcenter server. The minimum value is **1**.

**Object data indexing start and end times**

**Start Time**

All data modified after this date and time are extracted for indexing; data older than this date is not extracted. This value is only used during first-time indexing or re-indexing.

| | |
|---|---|
| | **End Time**<br><br>If selected, it specifies the end date for extracting data. Data modified after this date will not be extracted for indexing. This value is only used during first-time indexing or re-indexing.<br><br>If no end time is specified, all data modified from the start time to the present is indexed. |
| **Maximum Query Timespan** | Specifies the maximum span of a Teamcenter query in minutes. The maximum value is **50000**; the minimum value is **5000**; the default value is **20000**. |
| **Export Batch Size** | Specifies the maximum number of Teamcenter objects handled in one thread. The maximum value is **20000**; the minimum value is **1**; the default value is **1000**. |

10. Click **Save Component Settings**.

11. Complete configuration of any remaining components.

12. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

13. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see the *Deployment Center — Usage*.

14. After installing the Indexer, you must optimize instances of **TcFTSIndexer** by adjusting the maximum Teamcenter connections value, maximum query timespan, and export batch size as necessary.

## Install Teamcenter Artificial Intelligence Services using Deployment Center

Add the separately licensed Teamcenter Artificial Intelligence (AI) Chat application to your existing Teamcenter environment. Installation of Teamcenter AI Chat enables Teamcenter to connect with your third-party vector database and models. This allows you to identify, structure, and summarize your files and information stored in Teamcenter. Siemens Digital Industries Software support is provided only for this interface connection.

### Prerequisites

- Obtain a subscription to either **Amazon Web Services (AWS)** or **Microsoft Azure** as a third-party provider for the required vector database and models. Install and configure all the necessary services for either **AWS** or **Microsoft Azure**. For information on compatibility, refer to the *Interoperability Matrix*, which available from the **Downloads** section of Support Center.

- **AWS**

| Required Services | Definition | Installation and Configuration | Required Deployment Center Settings |
|---|---|---|---|
| **Amazon OpenSearch Serverless (aoss)** | Serves as the vector database for storing and querying large amounts of data efficiently. | See the *Amazon OpenSearch Service Developer Guide* in **AWS Documentation**. | Endpoint URL<br>IAM role |
| **AWS Bedrock** | Provides managed API-level access to various large language models (LLMs), including embedding models. | See the *Amazon Bedrock User Guide* in **AWS Documentation**. | Large language model type and name<br>LLM endpoint URL<br>Embedding model name<br>Embedding model endpoint URL<br>IAM role |
| **AWS Identity and Access Management (IAM)** | Manages access to **AWS** services and resources securely. | See the *AWS Identity and Access Management User Guide* in **AWS Documentation**. | Access key<br>Secret access key<br>Region<br>IAM role |

- **Microsoft Azure**

| Required Services | Definition | Installation and Configuration | Required Deployment Center Settings |
|---|---|---|---|
| **Azure AI Search Service** | Serves as the vector database for storing and querying large amounts of data efficiently. | See *Create a search service* in **Microsoft Azure AI Search Documentation**. | Region<br>Endpoint URL<br>Primary administrator API key |
| **Azure OpenAI Service** | Provides access to Open AI LLMs and embedding models. | See the **Azure OpenAI getting started** section in **Microsoft Azure OpenAI Service Documentation**. | Large language model name<br>LLM endpoint URL<br>LLM API key<br>Embedding model name<br>Embedding model endpoint URL<br>Embedding model API key |

- Review all **AWS** or **Microsoft Azure** security practices relating to AI, LLMs, and API keys.

- Install the **Indexing Engine** and **TcFTSIndexer**.

### Procedure

1. Log on to Deployment Center and select the environment to which you want to add Teamcenter AI Chat.

2. Go to the **Applications** tab. Click **Add or Remove Selected Applications** .

3. In the **Available Applications** panel, use the web browser search to find the **Teamcenter AI Chat** application. Select the application, and then click **Update Selected Applications**.

   Deployment Center automatically selects any additional dependent applications.

4. Go to the **Components** tab.

5. In the **Selected Components** list, select the **Teamcenter AI Microservices** and **Microservice Node** components.

6. In the **Teamcenter AI Microservices** panel, select an option in **AI Platform** and enter values for the following configuration parameters:

| AI Platform | Parameter | Description |
|---|---|---|
| **Amazon AWS** | **Access Key** | Specifies the authentication access key set up with Amazon AWS. |
| **Amazon AWS** | **Secret Key** | Specifies the authentication secret key set up with Amazon AWS. |
| **Amazon AWS** | **Region Name** | Specifies the Amazon AWS cloud deployment region. |
| **Amazon AWS** | **Model Settings** | **LLM Name** specifies the name of your large language model. This should be in the format *model type*/*model name*. For example, **anthropic/anthropic.claude-instant-v1**.<br>**Embedding Model Name** specifies the name of your embedding model.<br>**IAM Role** specifies the role set up during Amazon AWS configuration that has access to the models. |
| **Amazon AWS** | **Vector Database Settings** | **Endpoint URL** specifies the location where your vector database can receive requests.<br>**IAM Role** specifies the role set up during Amazon AWS configuration that has access to the database. |
| **Microsoft Azure** | **Region Name** | Specifies the Microsoft Azure cloud deployment region. |

| AI Platform | Parameter | Description |
|---|---|---|
| **Microsoft Azure** | **Model Settings** | **LLM Name** specifies the unique name of your deployed large language model.<br>**LLM Endpoint URL** specifies the location where your large language model can receive requests.<br>**LLM API Key** specifies the authentication key for the large language model.<br>**Embedding Model Name** specifies the unique name of your deployed embedding model.<br>**Embedding Endpoint URL** specifies the location where your embedding model can receive requests.<br>**Embedding API Key** specifies the authentication key for the embedding model. |
| **Microsoft Azure** | **Vector Database Settings** | **Endpoint URL** specifies the location where your vector database can receive requests.<br>**API Key** specifies the authentication key for the vector database. |
| **Self Managed** | **Model Settings** | **LLM Name** specifies the name of your large language model.<br>**LLM Endpoint URL** specifies the location where your large language model can receive requests. The port for this URL is **11434/v1**.<br>**Embedding Model Name** specifies the name of your embedding model.<br>**Embedding Endpoint URL** specifies the location where your embedding model can receive requests. The port for this URL is **11434** |
| **Self Managed** | **Vector Database Settings** | **Endpoint URL** specifies the location where your vector database can receive requests. The port for this URL is **9200**.<br>**User Name** and **Password** for the user name that has access to the vector database. |

To specify additional settings, click **Show all parameters**.

7. In the **Microservice Node** panel, verify that one instance of each of the following two microservices is running:

   - **Teamcenter Data Vectorizing Service**
   - **Teamcenter Language Model Invocation Service**

   To specify additional settings, click **Show all parameters**.

8. When you finish entering values for the components, click **Save Component Settings**.

9. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter the required parameters, and save the component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** tab is enabled.

10. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate the deployment scripts that you will use to update affected machines.

    When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

11. Locate deployment scripts, copy each script to the target machine, and then run each script on the target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

### Postrequisites

Configure Teamcenter AI Chat for your users.

Configure Natural Language Search for your users.

## Install Dispatcher

The Teamcenter *Dispatcher* is an asynchronous executor and load balancer of scheduled jobs. If you use Dispatcher, install the Dispatcher server and client as described in *Dispatcher — Deployment and Administration*. Then, install the following Dispatcher translators, which Active Workspace uses:

- **Active Content Structure Translator**

  Install this translator if you use Dispatcher-based indexing for structure data. This feature must be installed in the same environment as the Dispatcher server.

- **ReqMgmtWordToHtmlTrans** (optional)

  This translator converts requirements content that has been edited and saved in Microsoft Word from Teamcenter (stored as a full-text dataset), so that it can be viewed in the rich text editor in Active Workspace.

- **AsyncService** (optional)

  This translator provides asynchronous reporting and printing.

  To set up email notifications this translator uses, set the following preferences:

- **MAIL_OSMAIL_ACTIVATED** = **true**
- **MAIL_INTERNAL_MAIL_ACTIVATED** = **true**
- **MAIL_SERVER_CHARSET** = **ISO-8859-1**
- **MAIL_SERVER_NAME** = *mail-server-name*
- **MAIL_SERVER_PORT** = **25**
- **MAIL_SUBSCRIPTION_NOTIFY_SUB_GROUP_TOO** = **FALSE**
- **WEB_DEFAULT_SITE_SERVER** = *host*:*port*
- **WEB_DEFAULT_SITE_DEPLOYED_APP_NAME** = *Teamcenter-web-tier-application*

Note:

If your environment is distributed, Deployment Center may automatically add additional **business logic servers** as needed to support distributed components.

## Add a business logic server

A *business logic server* processes requests from Teamcenter clients, applying your business model and rules when accessing the Teamcenter database and files in Teamcenter volumes. It contains the Teamcenter server process (**TcServer**), which performs the core functions of Teamcenter.

The *corporate server* is the primary business logic server in your environment. Every Teamcenter environment contains one corporate server. But, a distributed environment can contain multiple additional business logic servers. (A single box environment supports only one business logic server, the corporate server.) Business logic server interaction with the Teamcenter database is managed by a server manager (the **Server Manager** component).

When you install a component that requires a Teamcenter server process on a machine other than the corporate server machine, Deployment Center automatically adds a **Business Logic Server** component on that machine. You can also optionally add a business logic server on a specific machine to further distribute server processes.

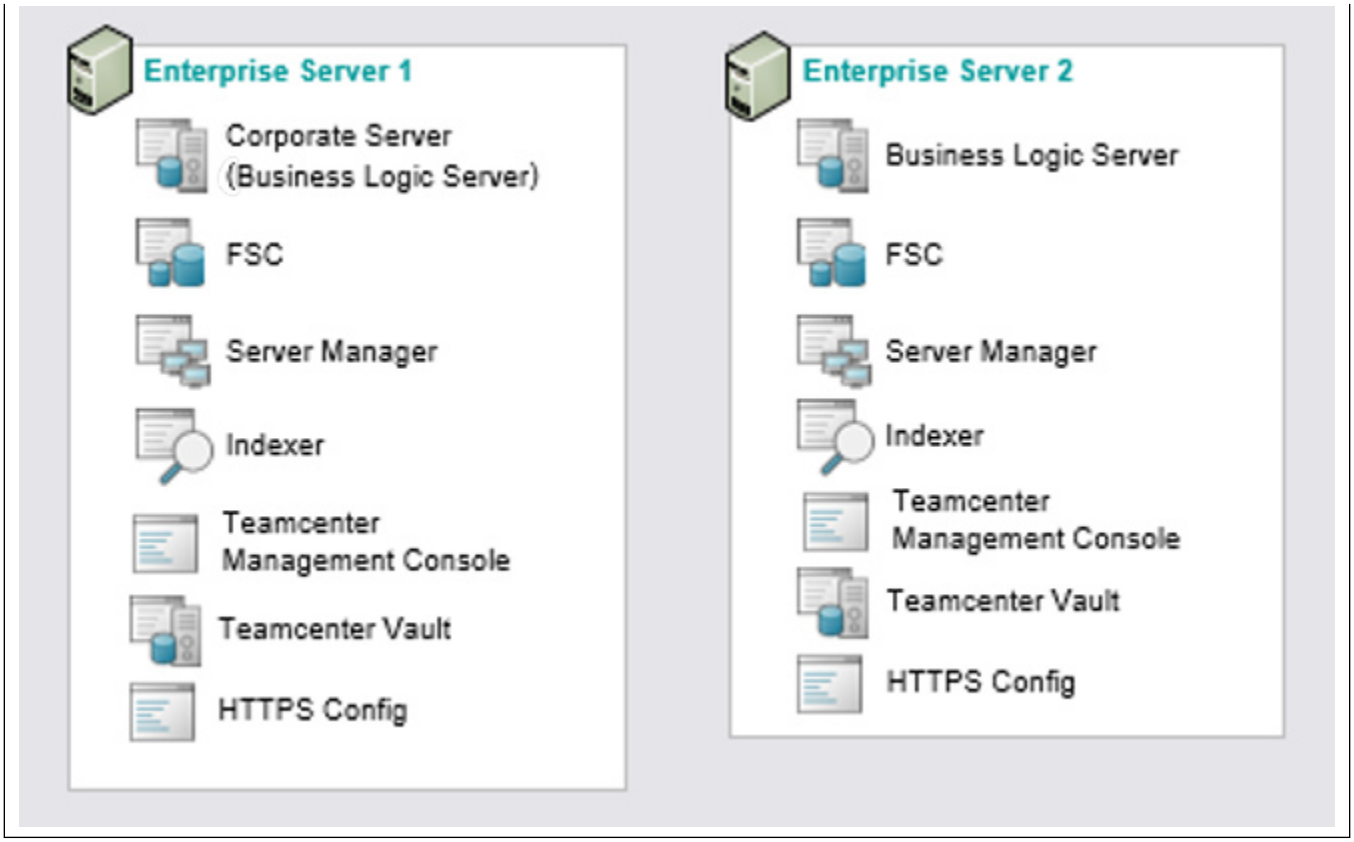



**Figure 3-1. Example: Corporate server and business logic server in a distributed environment**

A business logic server contains the same libraries as the corporate server and can share an existing Teamcenter application directory (*TC_ROOT*) or a Teamcenter data directory (*TC_DATA*) with another business logic server. Or, it can use its own local application and data directories.

To add a business logic server, go to the **Components** tab, click **Add a component to your environment** ⊕, select **Business Logic Server** in the **Available Components** list, and then click **Update Selected Components**.

Configure the business logic server:

1. Go to the **Components** tab. In the **Selected Components** list, select **Business Logic Server**.

2. Set the **Machine Name** and **OS** for the business logic server.

3. If you want to use a local Teamcenter application directory and data directory on the business logic server machine, type the Teamcenter installation path or accept the default **Teamcenter Installation Path** shown. Then, proceed to step 6.

4. If you want to use a shared Teamcenter application directory, select the **Shared TC_ROOT?** ☑ check box. Then, supply the path to the shared *TC_ROOT* location by choosing one of the following options:

- **Use Shared Teamcenter Root Path**

  Type the path to the *TC_ROOT* location on the target machine.

- **Use Environment Variable TC_ROOT_SHARED during deployment**

  Set the **TC_ROOT_SHARED** system environment variable on the target machine to path to the *TC_ROOT* location.

5. If you want to use a shared Teamcenter data directory, select the **Shared TC_DATA?** check box. Then, supply the path to the shared *TC_DATA* location by choosing one of the following options:

   - **Use Shared Teamcenter Data Path**

     Type the path to the *TC_DATA* location on the target machine.

   - **Use Environment Variable TC_DATA_SHARED during deployment**

     Set the **TC_DATA_SHARED** system environment variable on the target machine to path to the *TC_DATA* location.

6. Click **Save Component Settings**.

You can add a business logic server to a distributed environment only. It is not supported in single box environments.

## Visualization Server

### Visualization Server overview

The Visualization Server provides dynamic 3D and 2D visualization functionality to the Active Workspace client. If you do not use the 3D viewer or the 2D part of the universal viewer in Active Workspace, do not install the Visualization Server.

The Visualization Server comprises three components:

#### Visualization Server Manager

The Visualization Server Manager (VSM) starts and stops rendering processes as needed and streams visualization data to the Active Workspace client.

The Visualization Server Manager is required for any use of the 3D viewer or the 2D viewer part of the universal viewer in Active Workspace.

Siemens Digital Industries Software recommends that you install the Visualization Server Manager on a machine that does not have a Teamcenter corporate server.

### Visualization Server Pool Assigner

The Visualization Server Pool Assigner (VPA) manages Visualization Server Managers and routes users to an available VSM to open 3D documents.

Each Visualization Pool Assigner hosts two MXBeans that contain information about its current state: **Assigner** and **Assigner monitoring**. The MXBeans are located in the **Administer Assigner manager** folder.

Siemens Digital Industries Software recommends that you install the Visualization Server Manager on a machine that does not have a Teamcenter corporate server.

### Visualization Data Server **(optional)**

The Visualization Data Server (VDS) improves Visualization performance by caching visualization data close to the Visualization Server Manager.

The Visualization Data Server is required for using MMV feature in Active Workspace. Additionally, you need to index structure dataindex structure data for the product configurations that you want to view using MMV.

For appropriately indexed product configurations, the VDS performs the following to promote faster rendering and streaming to the Active Workspace client:

- Caches product structure
- Prepopulates JT files in the FCC
- Computes Massive Model Visualization (MMV) spatial hierarchies
- Provides bounding box validation

  You can use bounding box validation to suppress display of parts that fall outside a defined assembly box. This can help avoid assemblies opening zoomed out to accommodate errant parts located far outside the actual assembly. Bounding box validation can also limit a view to include only a preferred range of the assembly.

  Bounding box validation is described in *Visualization — Deployment and Administration* in the Active Workspace help library.

A single Visualization Data Server can support one or more Visualization Server Managers.

A Visualization Server Manager is required on the same host as the Visualization Data Server. A Visualization Data Server is required for implementation of MMV, but is otherwise optional.

## Choosing client-side or server-side rendering

### At a glance: client-side rendering versus server-side rendering

Client-side rendering (CSR) uses WebGL to leverage client-side graphics capabilities using the Active Workspace browser. Server-side rendering (SSR) does not require WebGL and is suited to larger structures. The following comparison may help you decide which option to use.

| Conditions | CSR | SSR |
|---|---|---|
| **Data size limit** | The data size limit is affected by browser memory, transfer time tolerance, and WebGL performance as the data size increases.<br>Using render acceleration can increase this limit significantly. | This option provides the highest data size limit, because the server has substantial CPU, RAM, and GPU resources. |
| **Load speed** | All data is streamed to the client.<br>Browser caching for client-side rendering is supported. | Best option for loading speed.<br>Data is localized to the render server. |
| **User experience interaction** | Best experience within the limits of WebGL performance.<br>All drawing and interactions are local.<br>Rendering is unaffected by network traffic, so is more responsive and less latency sensitive. | Good experience, especially with low latencies.<br>Better-to-best experience when working with significantly increased data sizes. |
| **Server cost per user** | Lower cost. No server-side graphics card is required.<br>Offloading rendering to clients means the system can support more users per server. However, the triangles of the model must be loaded onto the client machine before it can render. | This option has a higher cost, but it can be more cost effective than putting a high-end graphics device on every user's desk. |
| **Device support** | Devices that support WebGL and an HTML5 web browser. | Devices that support an HTML5 web browser. This option is necessary for devices that do *not* support WebGL. |

### Reserve slots on SSR servers for SSR users unless all CSR capacity is consumed

To optimize resource utilization, the Visualization Server Pool Assigner directs SSR users to SSR-capable servers, while diverting CSR users to servers that can support CSR users only. However, since SSR servers can also support CSR users, when all CSR servers are busy and the SSR servers still have capacity, you can use SSR servers to support CSR users. This provides flexibility within the enterprise while reserving SSR servers for users who need that resource.

To adjust or disable this behavior, you can contact Siemens Digital Industries Software support.

### Rendering 3D data

In Active Workspace, the **3D** viewer is displayed within the universal viewer area of the **Overview** tab for objects that have viewable attachments. The 3D viewer is also displayed in the **3D** tab, where you can explore 3D data (JT) associated with parts and assemblies. The render location setting applies to both viewer locations.

Visualization Server is required for visualizing 3D data in Active Workspace with CSR. However, the Visualization Server Manager can be installed on a server without a graphics card.

For better user experience and certain functionality to work, ensure the following:

- The client machine for CSR must have a valid graphics card.
- You must enable WebGL on the browser.
- For SSR, even the server must have a valid graphics card.

### Set default rendering method

To set the default rendering method for the 3D viewer and the universal viewer, set the value of the **AWV0ViewerRenderOption** Teamcenter preference to either of the following:

- For client-side rendering (default option): Set the value to **CSR**.
- For server-side rendering: Set the value to **SSR**.

End users can change the rendering method on the **Viewer Options** panel in Active Workspace.

### Additional settings for CSR

Ensure that you are not using integrated graphics, and perform the following steps to switch to your graphics card:

1. Open the NVIDIA Control Panel.
2. Click **3D Settings→Manage 3D Settings**.
3. Click the **Program Settings** tab.
4. From the list shown, select the program for which you want to choose a graphics card.
5. Select the preferred graphics processor from the list.

Alternatively, ensure that the GPU is used when running Google Chrome:

1. Open Windows settings (Windows key+I).

2. Search for graphics settings or GPU.

## Should I use MMV?

Massive Model Visualization (MMV) is a visualization technology that uses Visibility Guided Rendering (VGR) to increase performance and scalability when viewing extremely large 3D models, such as cars, airplanes, and ships. Models of this size typically consist of a massive amount of geometry arranged in a relatively compact space with a huge amount of internal geometry hidden behind the outer shell of the product. It can take hours to display such models in their entirety, because every piece of geometry in the model needs to be retrieved and processed, far exceeding the typical capability most hardware. MMV technology resolves this problem by leveraging VGR techniques to load only those parts that are required to render a given scene; parts that are not visible because they are occluded by other parts in the foreground are not loaded. As a result, large 3D models become visible in a fraction of the time previously required.

> Note:
>
> If a structure has more than 30,000 BOM lines, MMV is recommended. If a structure has more than 120,000 BOM lines, MMV rendering is required for scalability and performance.

Visualization of MMV data in the Active Workspace requires an MMV license. If the necessary license is not present, the full model loads as standard JT data.

A Visualization Data Server is required for implementing MMV but is otherwise optional.

To use the Visualization Data Server to compute Massive Model Visualization (MMV) spatial hierarchies of structures, you must do the following actions:

1. Apply the MMV index structure flag to the product configurations that you want to view using MMV.

2. Use the **bomindex_admin** utility to include the configurations in the list of structures to index.

The Visualization Data Server has a structure and JT pre-caching feature that can help improve visualization performance for structures not indexed for MMV. To use this feature:

1. Apply the VDS indexing flag for product configurations that will be viewed frequently but are not indexed for MMV.

2. Use the **bomindex_admin** utility to include the configurations in the list of structures to index.

## Visualization Server Manager

### Visualization Server Manager prerequisites

### Operating systems

The Visualization Server Manager (VSM) supports both large model visualization (LMV) and massive model visualization (MMV) on supported Microsoft Windows and Linux server platforms.

On a Linux machine without a GPU or without a supported level of OpenGL, client-side rendering is supported, but server-side rendering is not supported and fails to load.

For supported OS versions, see the Hardware and Software Certifications knowledge base article on Support Center.

### Server hardware and graphics cards

The following hardware is supported for VSM:

- **For server-side rendering:**

  Server class hardware certified by NVIDIA to support NVIDIA RTX 6000, RTX 8000, T4, A10, A40, GRID K1, K2, Tesla M60, or P40 graphics cards. Note that any server capable of supporting server-side rendering also supports client-side rendering.

- **For client-side rendering:**

  GPU hardware requirements for desktop Visualization applications (Lifecycle Visualization) are sufficient for client-side rendering.

  If no server-side rendering is needed, any web server class hardware is sufficient to support client-side rendering (CSR).

  The Visualization Server is required for visualizing 3D data in Active Workspace with client-side rendering. However, to use client-side rendering, you must install the Visualization Server Pool Assigner and VSM on a server without a graphics card.

Sizing of hardware should be appropriate to support intended data sizes and usage patterns. See *VSM hardware sizing* for more info about hardware sizing.

Windows Server versions supported with the VSM support a maximum of 8 GPUs, with certain exceptions. For example, on a Windows Server 2012 R2 machine with two NVIDIA GRID K1 cards, the legacy VGA device makes the fourth GPU on one card unavailable for use.

Active Workspace supports virtualized server-side rendering for **certain hardware and software combinations**.

NVIDIA usage requires NVIDIA virtual application licenses — one per concurrent user.

For information about server hardware compatible with supported NVIDIA GRID graphics cards, see **www.nvidia.com**.

### Virtualization

If you use only client-side rendering, the VSM can be virtualized.

If you use server-side rendering, the VSM must be installed on physical hardware, unless you follow a supported virtualization combination.

Active Workspace visualization supports virtualization for certain combinations of:

- Host OS and version
- Virtualization layer
- Guest OS and version
- NVIDIA GPU

For information about supported combinations, see the Graphics Card Certification Matrix in the Hardware and Software Certifications knowledge base article on Support Center: **https://support.sw.siemens.com**

For information about NVIDIA virtual GPU compatibility, see NVIDIA virtual GPU (vGPU) software documentation at **docs.nvidia.com**.

### VSM hardware sizing

Sizing of VSM hardware should allow for typical and maximum expected usage by considering the following factors:

- Expected numbers of concurrent Active Workspace visualization users
- Expected product data sizes
- CPU, RAM, VRAM and GPU resources consumed by expected product data

In general, a high end server with:

- A maximum number of CPU cores with processing speeds of 3.0 GHz or faster
- A minimum of 64 GB of RAM

- A minimum of 256 GB of disc space

In addition, a VSM that will support server-side rendering requires an NVIDIA GRID graphics card. For information about server hardware compatible with supported NVIDIA GRID graphics cards, see **www.nvidia.com**.

For additional guidance in sizing of VSM hardware, contact your field services professionals.

## Environment information

Make sure you know the following values. These are needed during installation of the VSM.

**Visualization Server Pool Assigner host and port**

These are defined in **Install the Visualization Server Pool Assigner**.

**Visualization Data Server host and port (if VDS is to be installed)**

These are defined in **Install the Visualization Data Server**.

**Host and port of FCC parents**

These are defined during Teamcenter installation.

## Linux machine configuration

Before you run the VSM on a Linux machine, perform the following steps:

1. Make sure the machine has the **required RPM package managers**.

2. Install the required fonts:

```
sudo yum install '*font*' --skip-broken
```

3. Make sure that Xserver is installed and running on **DISPLAY :0**. One way to verify this is to type the following command to determine whether the **X** or **Xorg** process is running:

```
ps -ef | grep "/usr/.*/X.*:0" | grep -v grep
```

The command returns output similar to the following:

```
>ps -ef | grep "/usr/.*/X.*:0" | grep -v grep
root      9533     1  0  2023 ?        00:00:00 sudo /usr/bin/Xorg :0
     -background none -verbose -auth /run/user/471/gdm/Xauthority
     -seat seat0 -listen tcp vt7
root      9534  9533  0  2023 tty7     00:25:42 /usr/bin/Xorg :0
     -background none -verbose -auth /run/user/471/gdm/Xauthority
     -seat seat0 -listen tcp vt7
```

The exact output from the **grep** command may vary depending on the configuration of your Linux environment, but the output *must* contain **/user/bin/Xorg :0** *or***/user/bin/X :0** where shown above. This indicates that the Xserver is running on **DISPLAY :0** and that your environment is supported.

If you reboot the system, you must restart Xserver.

> Note:
>
> The Xserver allows graphical user interfaces on a Linux system. When you install Linux, if you installed a minimal text-only environment, you must install Xserver according to your Linux Xserver installation and setup guide.
>
> If you installed a graphical environment, Xserver and Xorg should already be installed. The Xserver manages the display hardware to provide a graphical interface. To use X applications, install the Xorg X11 apps package.
>
> For information about configuring Xserver, see your Linux system configuration documentation.

4. After you verify that Xserver is running on **DISPLAY :0**, configure Xserver for offscreen and headless operation for use by the visualization server processes.

   - **Linux machine with no GPU:**

     After you verify that Xserver is running on **DISPLAY :0**, run the following commands:

     ```
     setenv DISPLAY :0
     xhost +
     ```

     You must run these commands after a reboot.

     The commands return output similar to the following:

     ```
     >setenv DISPLAY :0
     >xhost +
     access control disabled, clients can connect from any host
     ```

   - **Linux machine with GPU:**

     Set up the NVIDIA GPU on the Linux machine by running the **setup_xserver.sh** script provided in the Visualization Server Manager installation.

     a. Change to the *TC_ROOT***/vispoolmanager/jetty** directory.

     b. Type the following command:

     ```
     setup_xserver.sh default
     ```

The `default` parameter specifies to use the graphics card and bus id discovered by the script. If you do not specify this parameter, the script prompts you to confirm the card and bus id, and provides the opportunity to change these values if you want.

### Install the Visualization Server Manager

1. Log on to Deployment Center.
2. In the **Environments** list, select the environment to which you want to add the Visualization Server Manager (VSM), or click **Add Environment** to create a new environment.
3. In the **Software** tab, make sure the Teamcenter 2412 software kit is included in your environment.
4. Proceed to the **Applications** tab and then click **Add or Remove Selected Applications** .
5. In the **Available Applications** list, select **Visualization Extension**.

   This selection adds the **Visualization Server Manager** and **Visualization Pool Assigner** components to the environment.
6. Select **3D Visualization** or **Active workspace Visualization 2D Viewer** if you wish to add the **Viewer Administration** application, which provides an interface for monitoring Visualization Server components in Active Workspace.
7. Click **Update Selected Applications**.
8. Proceed to the **Components** tab.
9. In the **Selected Components** list, select **Visualization Server Manager**.
10. Enter configuration parameters for the VSM:

    a. If your environment type, which is specified in the **Options** tab, is **Distributed**, type the values for the **Machine Name** and **OS** for the machine deploying the VSM.

    Also, in the **Teamcenter Installation Path** box, type the path for the location where you will install Teamcenter software on the VSM machine.

    > Note:
    >
    > If your environment type is **Single Box**, then the **Machine Name**, **OS**, and **Teamcenter Installation Path** boxes are read-only and cannot be changed.

b. Under **Gateway Settings > File transfer protocol**, select whether to use secure protocol for VSM communication with the Active Workspace Gateway. If you select **https**, the VSM uses the HTTPS configuration specified in the **HTTPS Config** component.

   The **HTTPS Config** component is configured during creation of the Teamcenter environment.

c. If you want to change other default configuration parameters for the VSM, click **Show all parameters** and change values as necessary.

d. Click **Save Component Settings**.

11. In the **Components** tab, note any components which do not have a configuration status of **100%**. Enter or update configuration parameters until all components show a configuration status of **100%**.

12. Proceed to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines.

    When the script generation is complete, note any special instructions in the **Deploy Instructions** panel.

13. Locate the deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

### Start Visualization Server Manager

## Start Visualization Server Manager on Linux

To start the Visualization Server Manager (VSM) on a Linux machine, type the following command:

*TC_ROOT*/**vispoolmanager/run_servermgr.sh**

## Optional: Start VSM as a Linux daemon

Alternatively, you can start Visualization Server Manager as a daemon by running the **installservice.sh** command for each jetty server with admin permissions:

**installservice.sh***unique-service-name port user*

If you do not specify parameters, the script will run in interactive mode and prompt you for the information.

For example, from the *TC_ROOT/vispoolmanager/jetty* directory, type:

**sudo ./installservice.sh MyUniquePoolManager1 8090 MyUser**

*Uninstalling the Linux daemon:*

To *uninstall* the VSM daemon, run the **uninstallservice.sh** command for each jetty server with admin permissions:

**uninstallservice.sh** *service-name*

For example, from the *TC_ROOT/vispoolmanager/jetty* directory, type:

**sudo ./uninstallservice.sh MyUniquePoolManager1**

If you do not know the name of the service, search the *TC_ROOT/vispoolmanager/jetty/* directory or the */etc/systemd/system* directory for a file named *service-name.service*. The *service-name* is the unique service name you provided to the **installservice.sh** command.

### Start the Visualization Server Manager on Windows

1. Make sure the **FMS_HOME** environment variable is set as a system environment variable and not a user variable. Visualization Server Manager runs as a service only if **FMS_HOME** is a system environment variable.

2. Run the following file:

   *TC_ROOT***vispoolmanager\run_visservermgr.cmd**

   > Note:
   >
   > If Visualization Server Pool Assigner (VPA) is not running, Visualization Server Manager displays console messages until it finds the VPA. To avoid this, **start Visualization Server Pool Assigner** before you start Visualization Server Manager.

When running **run_visservermgr.cmd**, you can use Windows remote desktop connection to sign on to the machine on which Visualization Server Manager is installed if you have an NVIDIA card with a driver version of 340.66 or later. Other remote access products, such as VNC, can also be used.

After running **run_visservermgr.cmd**, you can lock the machine, but you must remain logged on. If you sign out, Visualization Server Manager is shut down.

Visualization Server Manager requires access to the graphics card. Therefore, it cannot run as a Windows service in server-side rendering mode. However, you can start Visualization Server Manager as a Windows service when you use client-side rendering exclusively.

### Optional: Configure automatic logon and restart on Windows

You can configure Windows to automatically log on and restart Visualization Server Manager in the event of a system reboot.

> Caution:
>
> Enabling automatic logon bypasses security. When Windows is configured to automatically log on, anyone with physical access to the machine can restart it and gain entry to the system. Use automatic logon *only* if the system is in a secure environment.

1. Open the Windows User Accounts dialog box:

   a. Press the Windows key+R to display the **Run** dialog box.

   b. In **Open**, type **netplwiz**, and then click **OK**.

2. In the **User Accounts** dialog box, select a user account from the list.

3. Clear the **Users must enter a user name and password to use this computer** check box.

4. Click **Apply**.

   The **Automatically sign in** dialog box appears.

5. In the **Password** and **Confirm Password** boxes, type the user's password.

6. Click **OK**.

   The specified user is automatically logged on when Windows starts.

7. Create a script or batch file to launch the Visualization Server Manager. Include the following command to lock the workstation:

   ```
   rundll32.exe user32.dll LockWorkStation
   ```

8. Create a new task with Windows Task Scheduler to run the script or batch file at log on.

### Optional: Start Visualization Server Manager as a Windows service

You can start Visualization Server Manager as a Windows service only when you are exclusively using client-side rendering. Windows services cannot access the graphics card, so this is not a suitable deployment for server-side rendering.

1. To install Visualization Server Manager as a Windows service, run the **installservice.bat** command:

```
installservice.bat "%JAVA_HOME%" "VSM-dir" pool-IDport
```

Replace *VSM-dir* with the path to the Visualization Server Manager's **jetty** directory. Replace *pool-ID* and *port* with the ID and port for the **VisPoolManager** service. The port must match the **VisPoolProxy.poolUrl** port in the **jettyservice.properties** file.

For example:

```
installservice.bat "%JAVA_HOME%" "%TC_ROOT%\vispoolmanager\jetty" vispool-A 8090
```

2. Locate the newly installed service named **Teamcenter VisServlet***pool-ID* in the list of Windows services.

3. Right-click the service name and choose **Properties**.

4. On the **Log On** tab, enter logon credentials for the domain user account under which the service runs.

   Visualization Server Manager requires an FMS client cache (FCC) to cache files. Use a dedicated account to run this service, not the **Local System** account.

   Windows attempts to run the service automatically by default. If the service is not already running, a problem may have occurred.

   If you set the service to start manually in its **Properties**, then you can click **Run** from the toolbar to start the service, or right-click the service in the **Services** window and choose **Start**.

   To stop the service, either click **Stop Service** on the toolbar, or right-click the service and choose **Stop**.

To uninstall the service, type **uninstallservice.bat "Teamcenter VisServlet** *pool-ID*.

### Test Visualization from the Active Workspace client interface

Before you begin the following procedure, make sure the Visualization Server Manager installation *and* the Active Workspace client installation tasks are complete.

You can test the Visualization Server by logging on with the Active Workspace interface and viewing Visualization data, for example, a JT file.

1. Ensure that the following are running:

   - Visualization Server Manager
   - Visualization Pool Assigner

- Active Workspace Gateway
- Web application server hosting the Teamcenter web tier application
- Teamcenter server manager
- Teamcenter database

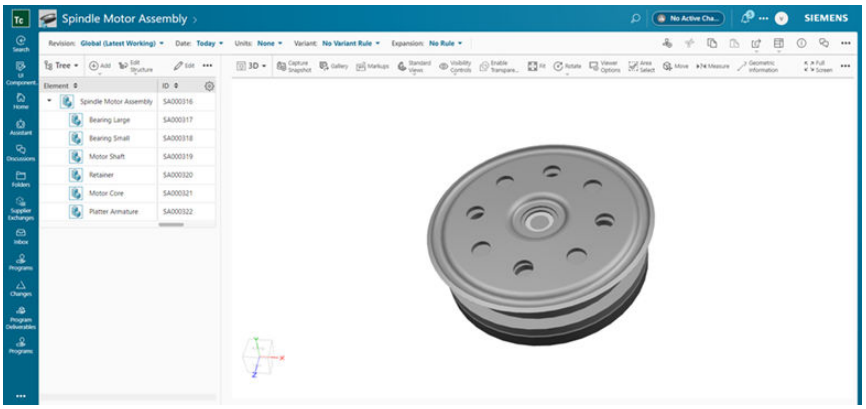
2. Open a supported web browser.
3. Open Active Workspace at the following URL:

   **http://***host***:***port*

   *host* is the machine running the Active Workspace Gateway.

   *port* is the port used by the Active Workspace Gateway.

4. Sign in with a valid user name and password.
5. Search for and open an object that has an attached JT file.
6. Click the **3D** tab to display the JT file.




### Configure the locale for Visualization Server Manager

You can configure the Active Workspace client to display the user interface in any of the supported Teamcenter locales. However, some visualization data, such as Product and Manufacturing Information (PMI), requires Visualization Server Manager (VSM) configured for the same locale as the information. For visualization data to display correctly in Active Workspace, you must have at least one VSM configured to run in each locale that you support. With a VSM in place to support each localization being used, visualization processes are then routed to the appropriate server based on locale.

VSMs can be configured to support the following languages:

| | | |
|---|---|---|
| Brazilian Portuguese | English | Korean |
| Chinese (Simplified) | French | Polish |
| Chinese (Traditional) | German | Spanish |
| Czech | Italian | Russian |
| French | Japanese | |

You can configure a VSM with any one of these languages. If you want to configure a cluster of VSMs to support more than one language, you need at least one VSM per language.

To change the language of a VSM, set the operating system (Windows or Linux) to the required language, location, and locale:

**For Windows systems**

1. Adjust the required language, location, and locale using the **Region** and **Language** options found in the Windows **Control Panel**.

2. Adjust the **Date and time formats**, the **Current location**, and the **Current language for non-Unicode programs** values.

3. Reboot the system after changing your Windows settings.

   When the VSM is started again, it inherits the new language configuration of the operating system.

**For Linux systems**

1. Run the following command to list all languages currently available on the machine: **locale -a**.

2. To configure the VSM to support a particular language, set the environment variables **LANG** or **LC_ALL** in the *jettyservice.properties* file.

   Example:

   To set the VSM to run using the German UTF8 locale, set these values in the *jettyservice.properties* file:

   - VisPoolProxy.envset.LANG = de_DE.utf8
   - VisPoolProxy.envset.LC_ALL = de_DE.utf8

> Note:
>
> Some Asian locales may require a restart of the Visualization Server to force the necessary fonts for the desired language to load correctly.

If all VSMs are configured to use the same language, all clients use the available language regardless of browser preferences.

Note that if you have a VSM system configured for two or more different languages, then Siemens Digital Industries Software highly recommends that at least one VSM be configured for English, even though this may require a minimum of three VSMs. When the server system is configured with multiple languages, if at least one VSM is configured for English, then the English locale is a default.

The following table shows the VSM system response to a visualization data request from a client when the client is not in one of the preconfigured languages.

| VSM system configured for two or more languages | Client is not in a preconfigured VSM language |
|---|---|
| VSM for English exists. | The data request is routed to an English VSM. |
| No VSM for English. | The data request is rejected. |

## Visualization Server Pool Assigner

### Visualization Server Pool Assigner prerequisites

#### Software

The Visualization Server Pool Assigner requires the following software:

- A supported Microsoft Windows Server operating system or Linux operating system on the Visualization Server Manager host.

  For supported versions, see the Hardware and Software Certifications knowledge base article on Support Center.

- The **Visualization Extension** Server Extensions feature on the corporate server and on any server that has Teamcenter Foundation installed.

#### Environment information

Make sure you know the following values. These are needed during installation of the Visualization Server Manager.

Visualization Server Pool Assigner host and port

| | |
|---|---|
| **Visualization Data Server host and port (if VDS is to be installed)** | These are defined in **Install the Visualization Data Server**. |
| **Host and port of FCC parents** | These are defined during Teamcenter installation. |

### Install the Visualization Server Pool Assigner

1. Log on to Deployment Center.
2. In the **Environments** list, select the environment to which you want to add the Visualization Server Pool Assigner (VPA), or click **Add Environment** to create a new environment.
3. In the **Software** tab, make sure the Teamcenter 2412 software kit is included in your environment.
4. Proceed to the **Applications** tab and then click **Add or Remove Selected Applications**.
5. In the **Available Applications** list, select **Visualization Extension**, and then click **Update Selected Applications**.

   This adds the **Visualization Server Manager** and **Visualization Pool Assigner** components to the environment.
6. Proceed to the **Components** tab.
7. In the **Selected Components** list, select **Visualization Pool Assigner**.
8. Enter the configuration parameters for the Visualization Pool Assigner (VPA):

   a. If your environment type, which is specified in the **Options** tab, is **Distributed**, type values for the **Machine Name** and **OS** for the machine on which you deploy the VPA.

   Also, in the **Teamcenter Installation Path** box, type the path in which to install Teamcenter software on the VPA machine.

   > Note:
   >
   > If your environment type is **Single Box**, the **Machine Name**, **OS**, and **Teamcenter Installation Path** boxes are read-only and cannot be changed.

b. Enter the following configuration parameters for the VPA:

| Value | Description |
|---|---|
| **File transfer protocol** | Specifies whether to use secure protocol for VPA communication with the Active Workspace Gateway. If you select **https**, the VPA uses the HTTPS configuration specified in the **HTTPS Config** component.<br><br>The **HTTPS Config** component is configured during creation of the Teamcenter environment. |
| **Vis Assigner Port** | Specifies the port used by the local Visualization Server Pool Assigner. |
| **Gateway Connection Port** | Specifies the port through which the Active Workspace Gateway connects to the Visualization Server Pool Assigner. The default value is **8089**. |
| **Gateway Vis Assigner URL** | Specifies the URL through which the Active Workspace Gateway accesses the VPA. This value is automatically based on the **Machine Name** and **Gateway Connection Port** values. It is read-only and cannot be directly changed. |

c. If you want to change the default configuration parameters for the VSM, click **Show all parameters** and change values as necessary.

d. Click **Save Component Settings**.

9. In the **Components** tab, note components whose configuration status is not **100%**. Enter or update configuration parameters until all components show a configuration status of **100%**.

10. Proceed to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines.

    When the script generation is complete, note any special instructions in the **Deploy Instructions** panel.

11. Locate the deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

### Start Visualization Server Pool Assigner

#### Linux systems

To start Visualization Server Pool Assigner (VPA) on a Linux machine, type the following command:

*TC_ROOT*\visassigner\run_assigner.sh

Alternatively, you can start the VPA as a daemon by running the **installservice.sh** command for each jetty server with admin permissions:

**installservice.sh** *unique-service-name port user*

If you do not specify parameters, the script runs in interactive mode and prompts you for the information.

For example, from the *TC_ROOT*\visassigner\jetty directory, type:

**sudo ./installservice.sh MyUniqueAssigner1 7780 MyUser**

**Windows systems**

To start VPA on a Windows machine, run the following file:

*TC_ROOT*\visassigner\run_visassigner.cmd

After running **run_visassigner.cmd**, you can lock the machine, but you must remain logged on. If you sign out, the VPA is shut down.

Alternatively, you can start VPA as a Windows service only when you are exclusively using client-side rendering. Windows services cannot access the graphics card, so this is not a suitable deployment for server-side rendering.

1. To install VPA as a Windows service, run the **installservice.bat** command:

   `installservice.bat "%JAVA_HOME%" "`*VPA-dir*`" `*assigner-IDport*

   Replace *VPA-dir* with the path to the VPA's *jetty* directory. Replace *assigner-ID* and *port* with the ID and port used by Active Workspace Gateway to connect to the Assigner.

   For example:

   ```
   installservice.bat "%JAVA_HOME%" "%TC_ROOT%\visassigner\jetty" VisAssigner-A
   8089
   ```

2. Locate the newly installed service named **Teamcenter VisServlet***assigner-ID* in the list of Windows services.

3. Right-click the service name and choose **Properties**.

4. On the **Log On** tab, enter the logon credentials for the domain user account under which the service runs.

Windows attempts to run the service automatically by default. If the service is not already running, a problem may have occurred.

If you set the service to start manually in its **Properties**, then you can click **Run** from the toolbar to start the service, or right-click the service in the **Services** window and choose **Start**.

To stop the service, either click **Stop Service** on the toolbar, or right-click the service and choose **Stop**.

To uninstall the service, type `uninstallservice.bat "Teamcenter VisServlet` *assigner-ID*`"`.

## Visualization Data Server (optional)

### Visualization Data Server prerequisites

#### Software

The Visualization Data Server requires the following software:

- A supported Microsoft Windows Server operating system or Linux operating system on the Visualization Server Manager host.

  For supported versions, see the Hardware and Software Certifications knowledge base article on Support Center.

- The **Visualization Extension** Server Extensions feature on the corporate server and on any server that has Teamcenter Foundation installed.

- A **Visualization Server Manager** installed on the Visualization Data Server host.

- An FMS client cache (FCC) component on the Visualization Data server host.

- Structure indexing configured on the Visualization Data server host.

  The Visualization Data Server uses the structure indexing infrastructure of Active Workspace to keep cached product structure up-to-date.

#### Hardware

- Graphics card: No requirements.

- Network: You must deploy the Visualization Data Server on a high speed LAN near the Visualization Server Manager.

- Memory: The Visualization Data Server host should have a minimum of 16 GB of RAM, but may require more.

> Note:
>
> **How to determine memory needed:**
>
> The amount of RAM needed depends on the number of structures to be indexed and their size.
>
> A rough rule of thumb is to count the number of lines in the unconfigured structure to be indexed and allow at least 2000 bytes per line. For example, if there are 1 million lines in the unconfigured product index, then 1 million * 2000 = 2 GB of RAM.
>
> If you are not sure of the size of the structures, Siemens Digital Industries Software recommends that you allow approximately 4 GB of RAM for each structure you are planning to cache in the Visualization Data Server. For example, if 4 structures are to be indexed, 16 GB of RAM is recommended.

### Environment information

You need to know the following values to install the Visualization Data Server:

- FCC parents
- Teamcenter web tier URL
- Host name and port for the Visualization Data Server

### Recommendations

Siemens Digital Industries Software recommends that you install the Visualization Data Server on a machine with the following:

- **Multiple processors**

  The Visualization Data Server is a multithreaded server program and is thus resource intensive; multiple processors are utilized if they are available. Standard server class machine hardware is sufficient.

- **FSC cache or FSC volume**

  If you deploy the Visualization Data Server remote (on a WAN) from the FSC volume, you should deploy an FSC cache on a LAN near or on the Visualization Data Server host machine.

- **Visualization Server Manager**

  For maximum performance, the Visualization Data server should be installed on the same machine as the Visualization Server Manager and should use the same cache.

A single Visualization Data Server can support one or more Visualization Server Managers.

### Install the Visualization Data Server

1. Log on to Deployment Center.
2. In the **Environments** list, select the environment to which you want to add the Visualization Data Server (VDS), or click **Add Environment** to create a new environment.
3. In the **Software** tab, make sure the Teamcenter 2412 software kit is included in your environment.
4. Proceed to the **Applications** tab, and then click **Add or Remove Selected Applications**.
5. In the **Available Applications** list, select **Visualization Extension**, and then click **Update Selected Applications**.
6. Proceed to the **Components** tab.
7. In the **Selected Components** list, select **Visualization Data Server**.
8. Enter configuration parameters for the VDS:

   a. If your environment type, which is specified in the **Options** tab, is **Distributed**, then type the values for the **Machine Name** and **OS** for the machine on which you deploy the VDS.

      Also, in the **Teamcenter Installation Path** box, type the path in which to install Teamcenter software on the VDS machine.

      > Note:
      >
      > If your environment type is **Single Box**, then the **Machine Name**, **OS**, and **Teamcenter Installation Path** boxes are read-only and cannot be changed.

   b. If you want to change the default configuration parameters for the VDS, click **Show all parameters** and change the values as necessary.

   c. Click **Save Component Settings**.

9. In the **Components** tab, note components whose configuration status is not **100%**. Enter or update configuration parameters until all components show a configuration status of **100%**.
10. Proceed to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines.

    When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

11. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

### MMV indexing data

If you use Massive Model Visualization (MMV), configure MMV indexing.

When structures using MMV rendering are indexed, the last valid indexed data is always retained. So, you can always see MMV indexed data; however, the data in a structure may be more recent.

When MMV data is being indexed it may use a backup system. It is recommended that the administrator retains interim files so when an error occurs, they can be analyzed to determine the issue. These two Teamcenter preferences can be used to control the output of the generated files:

- **MMV_keep_generated_files**

  Use this preference to preserve the generated files for further examination. You can specify when generated files are kept by using these values:

  **1**: Keep the generated files when an error occurs.

  **2**: Always keep the generated files.

  **3**: Never keep the generated files.

- **MMV_staging_directory**

  Use this preference to control the working directory to be used for the **tcxml2mmp** conversion process on the Teamcenter server. If this is not set, the default temporary directory is used as staging directory.

### Start Visualization Data Server

To start Visualization Data Server Manager, enter the following command:

**Windows systems:** *TC_ROOT***VisDataServer\Program\VisDataServer.exe**

**Linux systems:** *TC_ROOT*/**VisDataServer/bin/VisDataServer**

After the Visualization Data Server is started, it automatically detects and caches product configurations that have been indexed with the MMV flag. These cached product configurations are ready for fast visualization with the MMV technology.

For a product configuration is ready for MMV visualization, the following criteria must be met:

- The product configuration has been indexed.
- Visualization Data Server has detected, downloaded, and cached the structure.
- Visualization Data Server has prepopulated the FMS system.

If you attempt to visualize a product configuration that is not yet completely indexed and cached in the Visualization Data Server, the viewer uses the regular non-MMV mode by default. Changes in the product configuration need to be re-indexed and reread by the Visualization Data Server before they can be displayed by the viewer.

Additional configuration for the Visualization Data Server is available in the **etc/VisDataServer.properties** file. This includes detailed logging and fine tuning for other settings. If you make changes to the properties file, you need to restart Visualization Data Server.

### Optional: Start the Visualization Data Server as a Linux daemon

To install these services, run the **installservice.sh** located in the **VisDataServer/bin** folder. Run this command with administrator permissions:

**installservice.sh***unique-service-nameuserFMS_HOME*

For example, from the **VisDataServer/bin/** directory, type:

**sudo ./installservice.sh VDS MyUser /VIS/VisServer/FCC**

If you do not specify arguments, the script runs in interactive mode and prompts you for the required values.

To uninstall services, run the **uninstallservice.sh** script for each Jetty server. Run this command with administrator permissions:

**uninstallservice.sh***service-name*

For example, from the **VisDataServer/bin/** directory, type:

**sudo ./uninstallservice.sh VDS**

If you do not specify arguments, the script runs in interactive mode and prompts you for the required values.

If you do not remember the name of the service, find it using the following steps:

1. Change to the *VisDataServer/bin/* directory or */etc/systemd/system* directory.

2. Search for a file named *name.service*. The *name* in this file name is the *unique-service-name* you specified when you installed the service using **installservice.sh**.

### Optional: Start the Visualization Data Server as a Windows service

1. Make sure the **FMS_HOME** environment variable is set as a system environment variable, not a user variable. The VDS runs as a service only if **FMS_HOME** is a system environment variable.

2. Inspect the *VisDataServer.properties* file and make sure all file paths specified in it are full paths, not relative paths.

3. Open a Teamcenter command prompt and change to the root directory of the Visualization Data Server.

4. Install the Visualization Data Server as a Windows service by running the **VisDataServer.exe** command with the **/registerService** argument:

   **VisDataServer.exe /registerService /displayName=***name***/startup=***option*

   Replace *name* with a display name for the service. Replace *option* with **automatic** or **manual**.

   For example:

   ```
   VisDataServer.exe /registerService /displayName=VisDataServer /startup=automatic
   ```

   Optional additional arguments:

| Argument | Description | Example |
|---|---|---|
| **description** | Specifies a description for the service. | `/description="VDS for Active Workspace"` |
| **config** | Specifies a configuration file to load for the application. | `/config="VDSConfig.txt"` |

   After the service is successfully installed, Windows displays the following message:

   ```
   The application has been successfully registered as a service.
   ```

5. Configure the VDS service:

   a. In the Windows **Services** dialog box, locate the VDS service by the name you specified in the **displayName** attribute.

   b. Right-click the service name and choose **Properties**.

   c. In the **Log on** tab, enable the service logon with the following options:

- **Log on as**: Select **This account**, and then enter the domain and user name, for example, **myDomain\myName**.
- **Password**: Enter and confirm the password for the user account.

> Note:
>
> The VDS requires an FMS client cache (FCC) to cache files. Use a dedicated account to run this service, not the **Local System** account.

Windows attempts to run the service automatically by default. If the service is not already running when you open the Windows **Services** dialog box, the installation may have failed.

If you set the service to start manually, right-click the service name and choose **Start**. To stop the service, right-click the service name and choose **Stop**.

To uninstall the service, run the **VisDataServer.exe** utility with the **/unregisterService** argument.

### Rebuild VDS repository from scratch

As the VDS repository is updated via deltas containing incremental changes from Teamcenter that occur as design data evolves, the repository used to support MMV viewing may introduce errors. To reduce errors, a good practice is to periodically regenerate the VDS repository from scratch. The default threshold for this scratch rebuild is every 500 delta updates, but this value is configurable by an administrator. A full regeneration of the VDS repository can be set to occur more or less often, depending on the observed need.

To change the number of deltas that are processed before a scratch rebuild of the VDS repository, set the **MMV_delta_collection_accumulation_limit** Teamcenter preference to a value higher or lower than the default value of 500. This will change how often the VDS rebuilds its repository from scratch.

> Note:
>
> To manage the number of delta files that are to be deleted, use the **MMP_PERCENTAGE_OF_DELTA_TO_PURGE** preference. Its default value is 100, but based on your need, you can set it to any value from 10 to 100. Refer to the following table to understand preference value limits:

| If preference value is set to | then preference value processed is |
|---|---|
| <=10 | 10 |
| 10 < value < 100 | value |
| >=100 | 100 |

All delta files are deleted when the **MMP_PERCENTAGE_OF_DELTA_TO_PURGE** preference value is set to 100.

> Example:
>
> Set the **MMV_delta_collection_accumulation_limit** preference value to 500.
>
> In this case, since the value is set to 500, it becomes the maximum delta limit. After 500 deltas are processed, a completely new mmp file is created.
>
> To delete 100% of the old delta files in the MMV dataset, set the **MMP_PERCENTAGE_OF_DELTA_TO_PURGE** preference value to 100. If you want to delete only 10% of the old delta files in the MMV dataset, set the **MMP_PERCENTAGE_OF_DELTA_TO_PURGE** preference value to 10.

### Visualization Data Server status log settings

Configuration for the Visualization Data Server is available in the **etc/VisDataServer.properties** file. This includes detailed logging and fine tuning for other settings. If you make changes to the properties file, you must restart the Visualization Data Server.

Log information includes the status of all products hosted by the Visualization Data Server.

```
#
# Status logger settings. The status logger can be of help showing
the current indexing status
# and also the current and waiting task to be processed.
#
# The interval to generate the status log (see the "Interval"
documentation
 for more info).
StatusLogger.StatusInterval=120
# This will output the name of the top level (root) node.
StatusLogger.ShowRootName = true
# Shows the timestamp of the indexed product.
StatusLogger.ShowTimestamp = true
# Shows the available revision rules of indexed product.
StatusLogger.ShowRevRule = true
# Shows the status of the Spatial JTs.
StatusLogger.ShowSpatialJt = true
# If ShowSpatialJt is true, also shows the file path of the Spatial JTs.
StatusLogger.ShowSpatialJtPath = true
# If ShowSpatialJt is true and a Spatial JT is missing, the string will
be added
in from of the path.
# This can be used if a specific string is needed to search for a
missing
file (like using the grep utility).
StatusLogger.MissingSpatialJtMessage = (missing)
```

```
# Shows all the versions of a product instead of just the latest one.
StatusLogger.ShowAllVersions= false
# Shows the active tasks being processed.
StatusLogger.ShowActiveTasks=true
# Shows any waiting tasks to be processed.
StatusLogger.ShowWaitingTasks=true
```

# Install the Teamcenter web tier

## Install the .NET web tier application

### Configure Microsoft IIS for the .NET web tier

The Teamcenter .NET web tier is an alternative to the Teamcenter Java EE web tier. It supports four-tier Teamcenter deployments and does not require a Java EE application server.

The Teamcenter .NET web tier requires a supported Microsoft Windows Server operating system and also the following Microsoft components:

- Microsoft Internet Information Services (IIS)
- Microsoft .NET Framework

For required versions of these products, see the Hardware and Software Certifications knowledge base article on Support Center.

Before you install the .NET web tier, configure the required role services in Microsoft IIS on a Windows Server host. You can perform this from a command line *or* by using the Windows Server Manager.

#### Install role services from a command line

Open a Windows command prompt as an administrator and enter the following command in a single line:

```
dism.exe /enable-feature /all /online /featureName:IIS-CommonHttpFeatures
/featureName:IIS-DefaultDocument /featureName:IIS-DirectoryBrowsing
/featureName:IIS-HttpErrors /featureName:IIS-StaticContent
/featureName:IIS-HttpRedirect /featureName:IIS-HealthAndDiagnostics
/featureName:IIS-HttpLogging /featureName:IIS-LoggingLibraries
/featureName:IIS-RequestMonitor /featureName:IIS-HttpTracing
/featureName:IIS-Performance /featureName:IIS-HttpCompressionStatic
/featureName:IIS-HttpCompressionDynamic /featureName:IIS-Security
/featureName:IIS-RequestFiltering /featureName:IIS-BasicAuthentication
/featureName:IIS-ClientCertificateMappingAuthentication
/featureName:IIS-DigestAuthentication
/featureName:IIS-IISCertificateMappingAuthentication
/featureName:IIS-IPSecurity /featureName:IIS-URLAuthorization
/featureName:IIS-WindowsAuthentication
/featureName:IIS-ApplicationDevelopment
/featureName:IIS-NetFxExtensibility45 /featureName:IIS-ASP
```

```
/featureName:IIS-ASPNET45 /featureName:IIS-CGI
/featureName:IIS-ISAPIExtensions /featureName:IIS-ISAPIFilter
/featureName:IIS-ServerSideIncludes /featureName:IIS-WebServerManagementTools
/featureName:IIS-ManagementConsole
```

### Install role services using Windows Server Manager

Open the Windows Server Manager. Verify the **Web Server (IIS)** role is installed on your host. If this role is not installed, install it according to your operating system documentation.

In the Windows Server Manager, under the **Web Server (IIS)** role, install the following role services:

**Common HTTP Features**

- **Default Document**
- **Directory Browsing**
- **HTTP Errors**
- **Static Content**
- **HTTP Redirection**

> Caution:
>
> Do *not* install the **WebDav Publishing** role service.

**Health and Diagnostics**

- **HTTP Logging**
- **Logging Tools**
- **Request Monitor**
- **Tracing**

**Performance**

- **Static Content Compression**
- **Dynamic Content Compression**

**Security**

- **Request Filtering**
- **Basic Authentication**
- **Client Certificate Mapping Authentication**
- **Digest Authentication**
- **IIS Client Certificate Mapping Authentication**
- **IP and Domain Restrictions**
- **URL Authorization**
- **Windows Authentication**

**Application Development**

- **.NET Extensibility 4.*x***
- **ASP**

**ASP.NET 4.** *x*
**CGI**
**ISAPI Extensions**
**ISAPI Filters**
**Server Side Includes**

Install only the available **ASP.NET 4.***x* role services. Do not install ASP.NET 3.*x* role services.
**Management Tools**

**IIS Management Console**

### Install the .NET web tier

Before you install the .NET web tier, make sure you log on using an account with administrative privileges and that you have access to the Teamcenter software kit. Also, make sure your host has the required software and **is configured for the Teamcenter .NET web tier**.

This procedure assumes you have an existing Teamcenter environment. Make sure all the required Teamcenter software kits have been added to your software repository in Deployment Center

1. Log on to Deployment Center and select your Teamcenter environment.

2. In the **Options** tab, make sure your selected **Architecture Type** is **.NET**.

3. Proceed to the **Components** tab.

   If the **Selected Components** list does not include **Teamcenter Web Tier (.Net)**, add this component:

   a. Click **Add component to your environment** ⊕ to display the **Available Components** panel.

   b. Select **Teamcenter Web Tier (.Net)**, and then click **Update Selected Components**.

4. In the **Selected Components** list, select **Teamcenter Web Tier (.Net)**.

5. Enter values for the machine on which you install the .NET web tier:

   - **Single box**

     If your environment is a **single box** environment, the **Machine Name**, **OS**, and **Teamcenter Installation Path** values are inherited from the first component you configured in your environment. Changing these values will change them for other components in your environment.

   - **Distributed**

If your environment is a **distributed** environment, type the **Machine Name**, **OS**, and **Teamcenter Installation Path** for the machine on which you install the .NET web tier.

6. Enter the required values to configure the .NET web tier:

| Value | Description |
|---|---|
| **Protocol** | Specifies the protocol to use to connect to the web tier (**http** or **https**).<br>If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| **Virtual Directory Name** | Specifies the IIS virtual directory name for Teamcenter .NET web tier deployment. The default value is **tc**. |
| **Teamcenter Connection Name** | Specifies a name for the web tier connection. |
| **Tag** | Specifies a tag for the environment that can be used to filter the list of TCCS environments during logon. |

The **Teamcenter 4-tier URL** value is not directly editable, but is composed from the protocol, port, machine name, and port specified in other parameters for the **Teamcenter Web Tier (.Net)** component, for example, **http://myHost:80/tc**

If you want to specify additional settings for the Indexing Engine, click **Show all parameters** .

7. Click **Save Component Settings**.

8. Complete configuration of any remaining components.

9. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

10. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

After you **install the server manager** and the .NET web tier, complete the .NET web tier installation by launching the Teamcenter Management Console.

## Install the Java EE web tier

The Teamcenter Java EE web tier application provides communication between Teamcenter clients and the enterprise tier.

Before you install the Java EE web tier, make sure you install:

- A Teamcenter server and server manager.
- A supported Java EE application server and the Java Runtime Environment (JRE) on the web tier host.[2]

This procedure assumes you have an existing Teamcenter environment. Make sure all the required Teamcenter software kits have been added to your software repository in Deployment Center

1. Log on to Deployment Center and select your Teamcenter environment.
2. In the **Options** tab, make sure your selected **Architecture Type** is **Java EE**.
3. Proceed to the **Components** tab.

   If the **Selected Components** list does not include **Teamcenter Web Tier (Java EE)**, add this component:

   a. Click **Add component to your environment** ⊕ to display the **Available Components** panel.

   b. Select **Teamcenter Web Tier (Java EE)**, and then click **Update Selected Components**.
4. In the **Selected Components** list, select **Teamcenter Web Tier (Java EE)**.
5. Enter values for the machine on which you install the Java EE web tier:

   - **Single box**

     If your environment is a **single box** environment, the **Machine Name**, **OS**, and **Teamcenter Installation Path** values are inherited from the first component you configured in your environment. Changing these values will change them for other components in your environment.

   - **Distributed**

     If your environment is a **distributed** environment, type the **Machine Name**, **OS**, and **Teamcenter Installation Path** for the machine on which you install the Java EE web tier.
6. Enter the required values to configure the Java EE web tier:

| Value | Description |
|---|---|
| **Protocol** | Specifies the protocol to use to connect to the web tier (**http** or **https**). |

2 For information about supported application servers and Java versions, see the Hardware and Software Certifications knowledge base article on Support Center.

| Value | Description |
|---|---|
| | If you choose **https**, you must complete the configuration parameters in the **HTTPS Config** component. |
| **Teamcenter Application Name** | Specifies a name for the Teamcenter web tier web application. The default value is **tc**. |
| **Teamcenter Connection Name** | Specifies a name for the web tier connection. |
| **Web App Server Machine Name** | Specifies the name of the machine that runs the Java EE web application server. This the machine on which you deploy the Java EE web tier WAR file (typically **tc.war**). |
| **JMX RMI Port** | Specifies the JMX RMI port number for the web server. For example, type **8088** for the default server manager port or **8089** for the default web tier port. |
| **Tag** | Specifies a tag for the environment that can be used to filter the list of TCCS environments during logon. |

The **Teamcenter 4-tier URL** value is not directly editable, but is composed from the protocol, port, and machine name specified in other parameters for the **Teamcenter Web Tier (Java EE)** component, for example, **http://myHost:7001/tc**

If you want to specify additional settings for the component, click **Show all parameters** .

7. Click **Save Component Settings**.

8. Complete configuration of any remaining components.

9. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

10. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

11. Locate the Java EE web tier WAR file (typically **tc.war**) generated with the deploy scripts.

    Deploy the web application on a supported application server.[3]

3 *Web Application Deployment* provides Teamcenter web tier deployment procedures for several supported application servers.

## Install a volume server

By default, you can create volumes only on local disks, but if you want to write files to volumes residing on remote disks (shared across the network), you can create a stand-alone volume server.


1. Log on to Deployment Center and choose the environment to which you want to add a volume server.
2. Proceed to the **Components** tab.
3. Click **Add component to your environment** ⊕ to display the **Available Components** panel.
4. Select **Volume**, and then click **Update Selected Components**.
5. In the **Selected Components** list, select **Volume**.
6. In the **Volume** panel, enter the required values to configure the volume:

| Value | Description |
|---|---|
| **Instance** | Specifies a name for the volume instance. |
| **Volume ID** | Specifies the ID of the FMS server cache (FSC) for the volume. |
| **Path** | Specifies the path to the volume directory. |
| **Name** | Specifies the name of the volume directory. |
| **Host** | Specifies the host name of the host on which the FSC resides. |
| **Assign to FSC Server** | Denotes that the volume is assigned to an FSC server. The **FSC Server ID** is based on the machine name of the FSC Master and is not directly editable. |
| **Assign to File Store Group** | Denotes that the volume is assigned to the file store group. |

7. Click **Save Component Settings**.
8. Complete configuration of any remaining components.
9. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.
10. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

    For more information about running deployment scripts, see *Deployment Center — Usage*.

This procedure installs a single volume server. To configure multiple volume servers for load balancing, and other advanced FMS configuration, see *Teamcenter Administration*.

# 13. Installing optional applications

## Install the Business Modeler IDE

### Choose a Business Modeler IDE installation type

Several types of Business Modeler IDE installation are possible. All BMIDE installation types can be used to create, import, and modify a template project, and can generate a template package which can be deployed using TEM or Deployment Center.

An important difference among the installation types is whether and how the BMIDE connects to a Teamcenter site. A Teamcenter site connection is necessary for some tasks:

**Perform data exchanges, such as:**

- Synchronize the data model in a BMIDE template project with the Teamcenter server database.
- Live update non-schema data, such as lists of values (LOVs), from the BMIDE to a production server without shutting down the production server.
- Live deploy a template to a test Teamcenter server.
- Incorporate live update changes made to the production environment into a BMIDE standard template project.

**Create certain data model elements, such as:**

- Business object display rule
- Dynamic list of values
- Business context rule
- Item revision definition configuration (IRDC)
- System stamp configuration
- Subtype of AppInterface, and many others

Use the following general procedure for choosing a Business Modeler IDE installation type.

1. Ensure that the machine meets prerequisites for a BMIDE.

   > Caution:
   >
   > Do not install BMIDE on a production environment corporate server. Doing so could have unintended consequences, especially during Teamcenter upgrade.

2. Choose the BMIDE installation type that you want to perform.

| Installation type | Teamcenter connection type | Advantage | Limitation |
|---|---|---|---|
| **2-tier** | Two-tier environment via TCCS. | Allows live deployments even while a web tier is inactive or down for maintenance. | Requires local network access. |
| **4-tier** | Four-tier environment via HTTP server. | Allows remote access and live deployments. | Requires an active web tier. |
| **Standalone** | None | No requirement for or possibility of unintentional interaction with any Teamcenter site. | Cannot perform actions that require connection to a Teamcenter site. |

You may alternatively choose to add BMIDE functionality into your existing Eclipse environment. This consists of manually patching your Eclipse environment with BMIDE jar files. Doing so offers the advantage of allowing you to work on BMIDE templates within your existing custom Eclipse environment. However, adding BMIDE functionality into your existing Eclipse environment does not offer BMIDE functionality to perform actions that require connection to a Teamcenter site.

## Install the Business Modeler IDE using Deployment Center - connected

Before you install the Business Modeler IDE, make sure you log on using an account with administrative privileges and that you have access to the Teamcenter software kit.

This procedure assumes you have an existing Teamcenter environment. Make sure all the required Teamcenter software kits have been added to your software repository in Deployment Center

1. Log on to Deployment Center and select your Teamcenter environment.

2. Proceed to the **Components** tab.

   If the **Selected Components** list does not include **Business Modeler IDE** component type you want to install, add the component:

   a. Click **Add component to your environment** ⊕ to display the **Available Components** panel.

   b. Select the **Business Modeler IDE** component type you want to install (2 Tier or 4 Tier), and then click **Update Selected Components**.

3. In the **Selected Components** list, select **Business Modeler IDE** *installation type*.

4. Enter the required values to configure the Business Modeler IDE:

| Value | Description |
|---|---|
| **Enable Mass Client Deploy?** (2-Tier and 4-Tier) | Generates a deploy script that can be run on multiple machines. |
| **Machine Name** | Identifies the target deployment machine. The name is used in naming the deploy script.<br><br>• **Single box**<br><br>If your environment is a single box environment, the **Machine Name**, **OS**, and **Teamcenter Installation Path** values are inherited from the first component you configured in your environment. Changing these values will change them for other components in your environment.<br><br>• **Distributed**<br><br>If your environment is a distributed environment, type the **Machine Name**, **OS**, and **Teamcenter Installation Path** for the machine on which you install the Business Modeler IDE. |
| **OS** | Identifies the operating system on the target machine. |
| **Teamcenter Installation Path** | Path on the target machine for the installation. See notes in the description above for **Machine Name**. |
| **Java Development Kit Path** | Path on the target machine to the Java development kit. |
| **Templates and Clients** | Select the applications and clients whose templates you want to copy to the target machine for use in customization. The Teamcenter foundation template is included by default. |
| **Connection Port** and **Connection Name** (2-tier) | Specifies the port number and name to use for the 2-tier connection between the Business Modeler IDE and the Teamcenter server. |
| **Compress(gzip) Web Application server response** (4-tier) | Compresses traffic between the Business Modeler IDE and the Web Application server. |
| **Character Encoding type** (2-tier) | Specifies the character encoding type used by the database server on the target machine. |

By default, some parameters are not displayed because they generally do need to be changed from the default values. They can be displayed by clicking **Show all parameters** .

5. Click **Save Component Settings**.

6. Complete configuration of any remaining components.

7. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

8. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

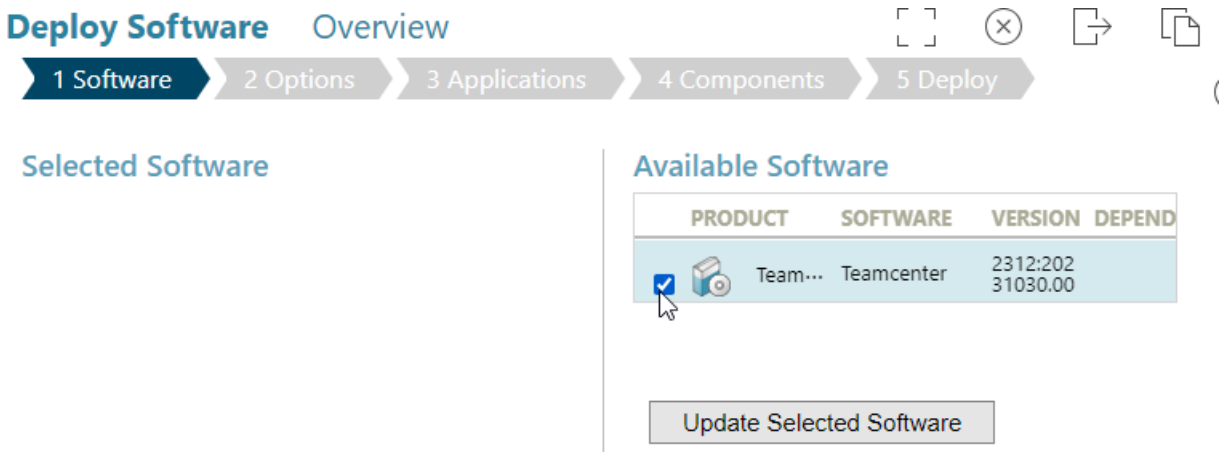
   For more information about running deployment scripts, see *Deployment Center — Usage*.

## Install the Business Modeler IDE using Deployment Center - standalone

Before you install the Business Modeler IDE, make sure you log on using an account with administrative privileges and that you have access to the Teamcenter software kit.

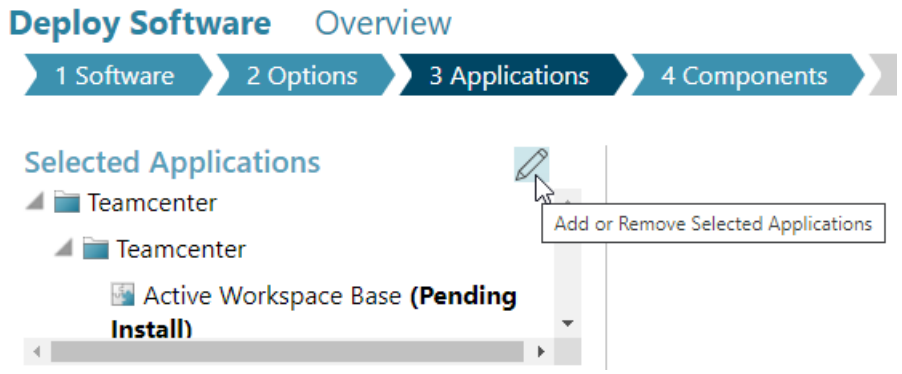
1. Log on to Deployment Center and create a new environment.


2. In the **Software** tab, add the Teamcenter software.

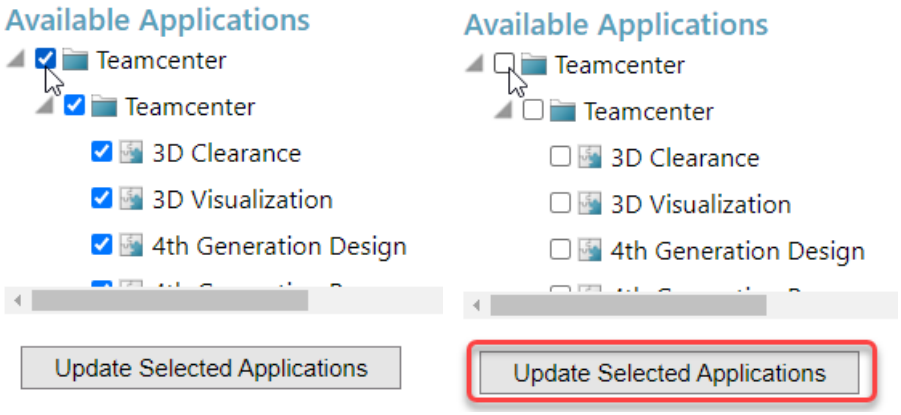



3. In the **Options** tab, accept the defaults (**Java EE** architecture, **Single Box** environment, **Local** architecture).

4. In the **Applications** tab, click **Add or Remove Selected Applications**.

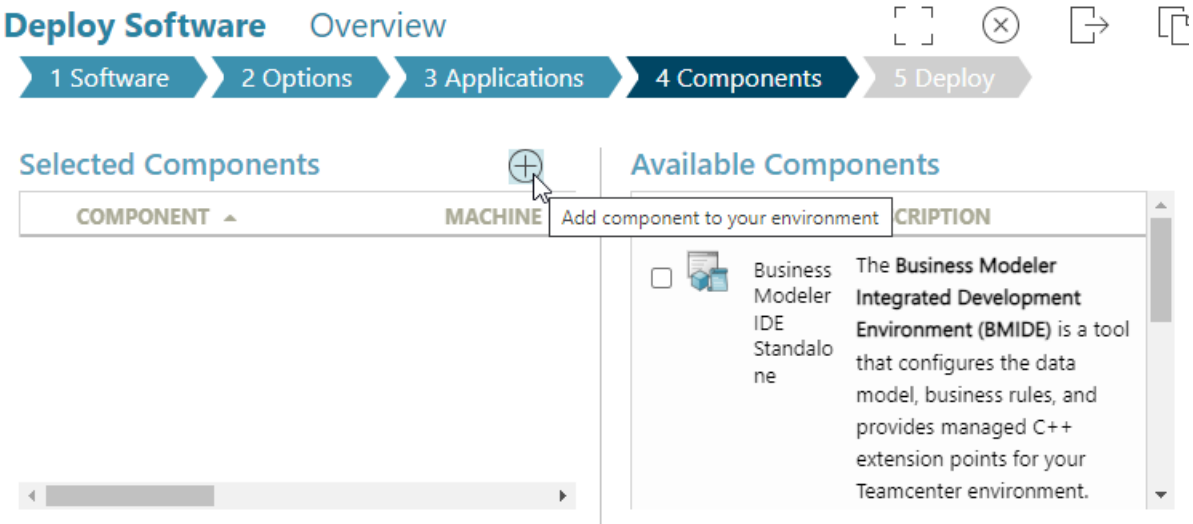



5. In the **Available Applications** list, select and then deselect the **Teamcenter** application group (this performs a select all/deselect all action), then click **Update Selected Applications**.



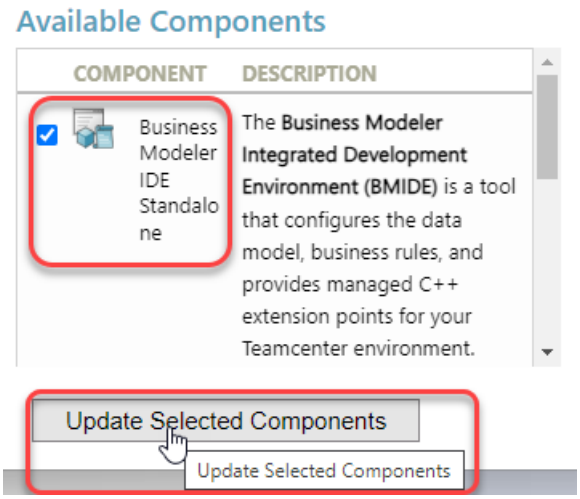

This clears the **Selected Applications** list.

6. In the **Components** tab, click **Add component to your environment** ⊕.

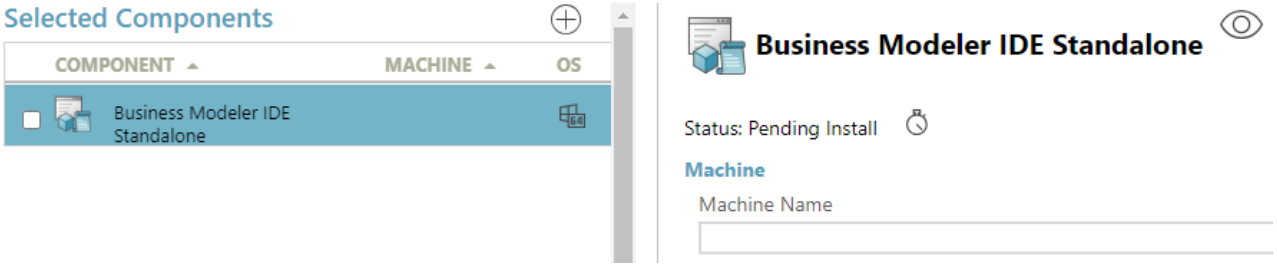


**Business Modeler IDE Standalone** is the only available component.

7. Select it and click **Update Selected Components**.



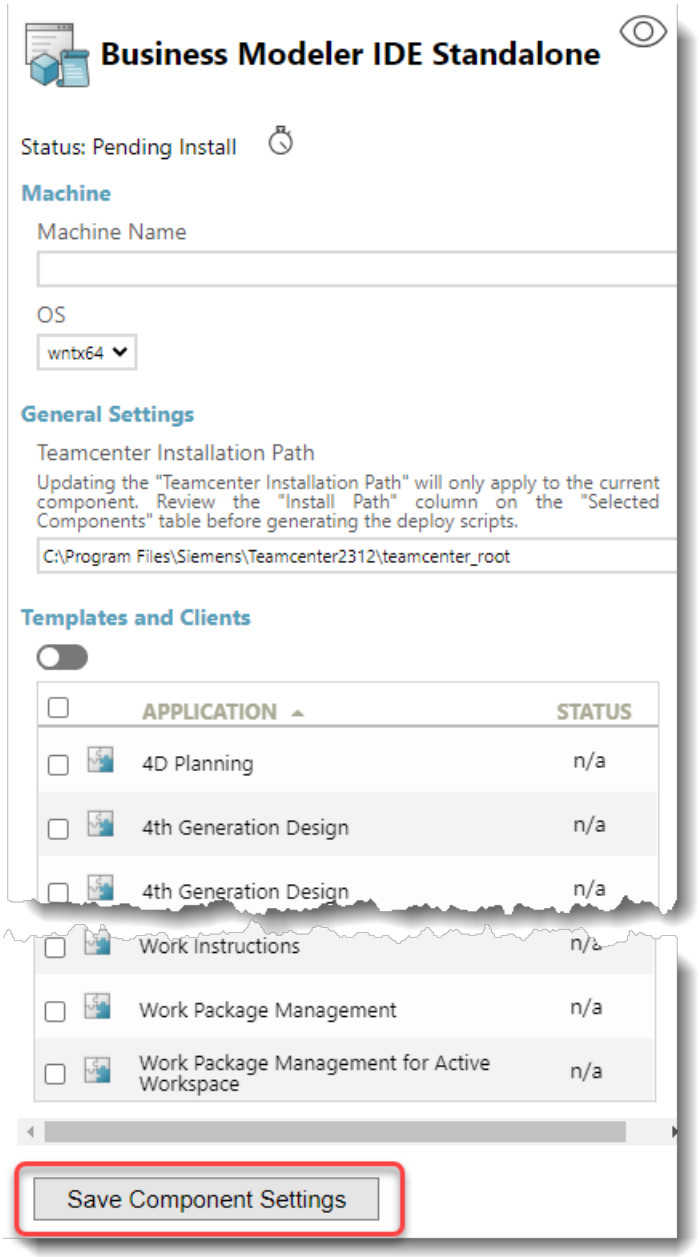

8. In the **Selected Components** panel, click **Business Modeler IDE Standalone** to view the parameters.




9. Enter parameters for the standalone Business Modeler IDE and then click **Save Component Settings**.

| Value | Description |
|---|---|
| **Machine Name** | Identifies the target deployment machine. The name is used in naming the deploy script. |
| **OS** | Identifies the operating system on the target machine. |
| **Teamcenter Installation Path** | Path on the target machine for the installation. |
| **Java Development Kit Path** | Path on the target machine to the Java development kit. |

| Value | Description |
|---|---|
| | By default, this parameter is not displayed. It can be displayed by clicking **Show all parameters** . |
| **Templates and Clients** | Select the applications and clients whose templates you want to copy to the target machine for use in customization. The Teamcenter foundation template is included by default. |




10. In the **Deploy** tab, click **Generate Install Scripts**.

11. Follow the **Deploy Instructions** to deploy the standalone BMIDE.

For more information about running deployment scripts, see *Deployment Center — Usage*.

## Allocate memory to the Business Modeler IDE

Allocate memory to the Business Modeler IDE so that it has enough to launch and run.

If you perform live updates, you must have a minimum of 2 GB of RAM on the system running the Business Modeler IDE to allow for other processes.

You can allocate memory in the following ways:

- **BusinessModelerIDE.ini** file

  To increase the memory allocated to the Business Modeler IDE, open the *install-location***bmide\client\BusinessModelerIDE.ini** file and change the **-Xmx1024M** value to a higher number to allocate maximum Java heap size. For example, if you have 2 GB available to dedicate for this purpose, set the value to **-Xmx2048M**. Do this only if your machine has the available memory.

  The **Xms** value in this file sets the initial Java heap size, and the **Xmx** value sets the maximum Java heap size.

- **BMIDE_SCRIPT_ARGS** environment variable

  To allocate the memory required by scripts during installation, update, or load of templates with large data models, create a **BMIDE_SCRIPT_ARGS** environment variable. Set the **BMIDE_SCRIPT_ARGS** variable to **-Xmx1024M** to allocate 1 GB of RAM to the Business Modeler IDE scripts. If your system has more memory that you can allocate to the Business Modeler IDE, you can set the value higher.

If you are running the Business Modeler IDE in an Eclipse environment, run the following command to increase virtual memory to 2 GB:

```
eclipse.exe -vmargs -Xmx2048M
```

Caution:

Java standards require that no more than 25 percent of total RAM be allocated to virtual memory. If the amount allocated to the Business Modeler IDE is higher than 25 percent of total RAM, then memory disk swapping occurs, with possible performance degradation.

If you set the **Xmx** value to a higher value than the RAM your system has, you may get the following error when you launch the Business Modeler IDE:

```
Could not create the Java virtual machine.
```

Set the **Xmx** value to a setting that your system supports, in both the **BMIDE_SCRIPT_ARGS** environment variable and the **BusinessModelerIDE.ini** file.

### Start the Business Modeler IDE

Start a Business Modeler IDE in one of several ways, depending on the installation type:

| Installation type | Platform | Procedure to start Business Modeler IDE |
|---|---|---|
| BMIDE Standalone, 2-tier, or 4-tier | Windows | Click the **Start** button and choose **All Programs**>**Teamcenter** [*version*]>**Business Modeler IDE**. This runs the **bmide.bat** file. |
| | Linux | Run the **bmide.sh** file in the *install-location*/**bmide/client** directory. |
| Eclipse environment to which BMIDE plug-ins have been added | Windows | Navigate to the directory where Eclipse is installed and execute the **Eclipse.exe** command.<br>`Eclipse.exe -vmargs -Xmx2024M`<br>To ensure that you have enough memory to run Eclipse, run the command with a virtual memory argument. In the example, the argument increases virtual memory to 2 GB. |
| | Linux | Navigate to the directory where Eclipse is installed and execute the **Eclipse** command.<br>`Eclipse -vmargs -Xmx2024M` |

For BMIDE operations that require connection to the Teamcenter server, users of the BMIDE must be members of the Teamcenter database administrators (**dba**) group. To add a user to the **dba** group, in the Teamcenter rich client use the Organization perspective.

If a perspective fails to open, it could be that not enough memory is allocated to the Business Modeler IDE.

## Installing custom software

### Deploy Business Modeler IDE packages

Users can generate a Business Modeler IDE template package in Teamcenter 11.3 or later that can be deployed to Teamcenter environments using either Deployment Center or Teamcenter Environment Manager (TEM). This consolidated output directory contains templates, libraries, and deployment configuration files.

To deploy a Business Modeler IDE template package, obtain the directory of the template package output generated by the Business Modeler IDE. Place the Business Modeler IDE output directory in the *software* subdirectory of the Deployment Center repository.

To ensure you have a supported template package, check:

- Directory naming convention

*template-internal-name_OS_template-version_build-version_YYYY_MM_DD_HH-MM-SS*

An optional template version may be assigned by the Business Modeler IDE user to track the versions of a template package. If the Business Modeler IDE user assigns a build number, the template is in development. The build version tracks iterative testing before the template is ready for production. Template versions and build versions are expressed as integers separated by periods, up to four places.

- *artifacts* subdirectory

  Contains the template software ZIP files for deployment.

- *dc_contributions* subdirectory

  Contains the template bundle information (called packages) for deployment by Deployment Center. If you use TEM, this directory is ignored.

- *tem_contributions* subdirectory

  Contains the template bundle information for deployment by TEM. If you use Deployment Center, this directory is ignored.

- **media_teamcenter_***template-package-name***.xml** file

  Provides the application names to both TEM and Deployment Center for deployment.

The Deployment Center repository displays **Dependencies** as specified within Business Modeler IDE packages using package IDs.

For information on creating and updating Business Modeler IDE packages, refer to the Business Modeler IDE documentation included with Teamcenter.

# Part III: Deploy the Teamcenter Environment


When you have satisfactorily configured and validated your Teamcenter test environment, you are ready to deploy to your environment as a production environment.

When you make your Teamcenter environment with Active Workspace available to users, you may want to explore options for large-scale deployment of clients to connect to your environment.

For information about deploying to a production environment and other deployment options with Deployment Center, see *Deployment Center — Usage*.

Also, see the *Teamcenter Deployment Reference Architecture*, available on Support Center, for further guidance and examples for development, test, and production environments.

# 14. Installing the Security Services Session Agent

## Install the Teamcenter Security Services Session Agent

The **Teamcenter Security Service Session Agent** provides authentication and single sign-on capability for Teamcenter desktop based clients and integrations.

This procedure assumes you have an existing Teamcenter environment with the required Teamcenter software kits in your software repository in Deployment Center.

1. Log on to Deployment Center and select your Teamcenter environment.

2. Proceed to the **Components** tab and click **Add component to your environment** ⊕ to display the **Available Components** panel.

3. Select **Teamcenter Security Service Session Agent**, and then click **Update Selected Components**.

4. In the **Selected Components** list, select **Teamcenter Security Service Session Agent** and then enter parameters for this component:

    a. If you configured the Session Agent in a **Global** infrastructure environment, you can import that component into your **Local** infrastructure environment.

    If you have no **Global** infrastructure environments, skip this step and proceed to step **b**.

    If you want to import a Teamcenter Security Services Session Agent configuration, perform the following steps:

    A. Select the **Do you want to import 'Teamcenter Security Service Session Agent' from other environments?** ☑ check box.

    Deployment Center displays a table of environments that contain a Teamcenter Security Services Session Agent component that can be shared to your environment.

    B. Select the environment from the list that contains the Session Agent component you want to import. Then, click **Save Component Settings**.

    The **Teamcenter Security Service Session Agent** component is fully configured (**100%** complete).

    C. Proceed to step **5**.

b. Enter parameter values as appropriate for your environment type:

- **Single box**

  All required parameters are supplied by existing components in your environment.

- **Distributed**

  Enter the required parameters below.

| Parameter | Description |
|---|---|
| **Enable Mass Client Deploy?** | Specifies you want to generate a deployment script that can be run on multiple client machines. If you select this check box, enter an identifier for the mass client instance in the **Instance Name** box. |
| **Machine Name** | Enter the name of the machine on which you want to install the rich client.<br>This box is displayed if **Enable Mass Client Deploy?** is *not* selected. |
| **OS** | Specifies the operating system of the machine on which you install the rich client. |
| **Teamcenter Installation Path** | Specifies the path in which to install the rich client on the target machine. Accept the default path shown, or type a different path. |
| **Install XML-RPC libraries** | If you have Teamcenter client applications released with Teamcenter 13.2 or earlier that you are not yet updating, select the **Install XML-RPC libraries** ☑ check box. This option ensures Security Services compatibility with earlier Teamcenter versions. |

c. Click **Save Component Settings** to submit the Session Agent configuration values.

a. Complete configuration of any remaining components.

5. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

6. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

# Configure the Session Agent

### Sharing an instance of the Session Agent

A local administrator can install the Session Agent in a common location, and that instance can be shared among multiple users. If an administrator has already installed the Session Agent on your client, then set the **TCSSO_SESSION_AGENT_PATH** user environment variable to the location of the Session Agent installation.

### Uninstalling the Session Agent

On Windows systems, you can uninstall the Session Agent from the Windows installed programs list. In the list, it is named **Teamcenter Security Services Session Agent**.

## Enabling digital signature support in the Session Agent

Previously, digital signature functions in Teamcenter (including digitally signing Teamcenter objects as well as digital signing for Workflow tasks) were supported through an ActiveX plugin installed on the client. Because ActiveX is no longer supported, the client-side processing for Teamcenter digital signatures has been moved to the Security Services Session Agent.

Digital signature enablement and configuration are supported only in Deployment Center, not in Teamcenter Environment Manager. Digital signature functions are also currently supported only on Windows clients.

### Add the Digital Signature Application

1. In Deployment Center, select your Teamcenter environment.
2. Select the **Applications** tab.
3. Click **Add or Remove Selected Applications** .
4. In the **Available Applications** list, under **Teamcenter→ Foundation**, select **Digital Signature**.

   > Note:
   >
   > The **Digital Signature** application is different from the **Digital Signatures** (with an **s**) application under **Teamcenter→ Active Workspace**.

5. Click **Update Selected Applications**.

### Configure Digital Signature settings in the Corporate Server

The settings required for digital signature support are stored in two places:

1. Server side: In the Teamcenter corporate server environment.

2. Client side: In the Teamcenter Security Services Session Agent environment.

The corresponding settings between these environments must match. Therefore, the settings have a single configuration point within Deployment Center to avoid a mismatch.

1. In Deployment Center, select the **Components** tab.

2. In the **Selected Components** list, select **Corporate Server**.

3. In the **Corporate Server** component settings, locate **Digital Signature Settings**.

4. Enter the required configuration parameters for digital signature support:

| Parameter | Description |
|---|---|
| **Port** | Specifies the port on which the Session Agent listens for digital signature requests. |
| **HMAC Secret** | Specifies a string secret that will be used to generate the hash-based message authentication code (HMAC) that secures digital signature communication with the Session Agent. |

Siemens Digital Industries Software recommends that the HMAC secret be randomly generated. Once configured, the secret will not need to be remembered by the administrator.

### Configure Digital Signature settings in Session Agent

1. In the **Components** tab, in the **Selected Components** list, select **Teamcenter Security Service Session Agent**.

   If this component is not in the **Selected Components** list, click **Add component to your environment** , and then add the component.

2. In the **Teamcenter Security Service Session Agent** component settings, enter the required configuration parameters for digital signature support:

| Parameter | Description |
|---|---|
| **Enable Digital Signature Functions** | Select this check box to configure the Session Agent to enable the digital signature functions. If enabled, the Session Agent process on the client machine will open an additional HTTP listener at the port specified in the **Corporate Server** component. If this option is disabled, the Session Agent process does not open the additional listener. |

| Parameter | Description |
| --- | --- |
| | Note:<br>Some Teamcenter clients in a Teamcenter environment may require support for digital signature functions while other clients do not. In that case, you can add multiple Session Agent components to the environment with the **Enable Digital Signature Functions** option set appropriately for each set of clients. |
| **CORS Whitelist** | Select this check box to define the list of origin URLs that will be returned in the Access-Control-Allow-Origins HTTP Header on responses sent by the digital signature endpoints in the Session Agent. This must include the Active Workspace Gateway URL. If this is not set correctly, the browser does not process the responses from the digital signature endpoints.<br>Example:<br>**https://MyActiveWorkspaceGatewayHost:3000**<br>Note:<br>Remember that the **Port** and **HMAC Secret** values are not available in the Session Agent parameters because they are referenced from the settings in the **Corporate Server** component. |

### Generate and Run the Deploy Scripts

After the **Corporate Server** and **Teamcenter Security Service Session Agent** components are configured for digital signature support, you can generate the install scripts.

In production deployments, the Session Agent is typically deployed using the **Enable Mass Client Deploy** option. This means that Deployment Center generates one install script for the server environment and one for each client machine. If you change the digital signature settings after the initial deployment, you must regenerate and redeploy the server script and client scripts to keep the settings synchronized.

# 15. Install the Active Workspace Launcher on a client machine

The Active Workspace Launcher application helps you open Microsoft Office and PDF files from Active Workspace to their native applications on the client machine. The Active Workspace Launcher application also opens the appropriate Office application when you open an attachment in Active Workspace.

> Note:
>
> Kerberos authentication is *not* supported with Client for Office.

## Prerequisites

1. Install Microsoft Office and Adobe Reader on the client machine.

   For supported versions, see the **Software Certifications Matrix** on Support Center.

2. Depending on your needs, install Teamcenter Client for Microsoft Office and/or Teamcenter Extensions for Microsoft Office as described in *Microsoft Office Integration With Teamcenter* in the Teamcenter documentation.

3. If you want to host Active Workspace within Client for Office, set Active Workspace hosting preferences as described in *Microsoft Office Integration With Teamcenter*.

## Procedure

1. In the Teamcenter 2412 software kit, locate the **wntx64\additional_applications\tcclientapplauncher\tcclientapplauncher.zip** file. Expand this file to a local directory.

2. Right-click the **setup.exe** program icon and choose **Run as administrator** to launch the Active Workspace Launcher installation wizard.

3. Proceed to the **Ready to Install the Program** dialog box, and then click **Install** to install the Active Workspace Launcher.

4. When the installation is complete, click **Finish** to close the installation wizard.

5. To enable the editing of requirements in Active Workspace, you must perform additional setup tasks.

### Install Active Workspace Launcher silently

Alternatively, you can install the Active Workspace Launcher silently, without user interaction:

1. To generate a silent installation file, type the following command in a command prompt:

   ```
   setup.exe /r /f1"path\tclauncher.iss"
   ```

   For example, to generate a silent file in the **c:\temp** folder, enter the following command:

   ```
   setup.exe /r /f1"c:\temp\tclauncher.iss"
   ```

   Do not include a space between the **f1** argument and the path that follows it. The path must be enclosed in double quotation marks (**" "**) as shown.

2. To install the Active Workspace launcher silently on another system, type the following command:

   ```
   setup.exe /s /f1"c:\tclauncher.iss"
   ```

### Troubleshoot the Active Workspace Launcher installation

If Microsoft Office applications fail to launch when opening an attachment, the **.awoai** file may be associated with Microsoft Word instead of the Active Workspace Launcher. To resolve this, perform one of the following tasks:

- Uninstall and reinstall the Teamcenter Active Workspace Launcher and try again.
- In the **HKEY_CLASSES_ROOT\awoai_auto_file\shell\open\command** registry entry, ensure the **.awoai** file extension is correctly associated with the **TcClientApplauncher.exe** command. For example, the key value should be similar to the following:

  **@="\"C:\\Program Files (x86)\\Siemens\\Teamcenter\\WSLauncheri\ \TcClientAppLauncher.exe\" \"%1\""**

# 16. Verify Active Workspace installation

To verify the Active Workspace installation is complete and successful, open the Active Workspace URL in a web browser:

**http://***host***:***port*

Replace *host* and *port* with the host and port of the Active Workspace Gateway.

For example:

**http://myhost:3000**

In the Active Workspace logon screen, enter the user name and password for the Teamcenter administrative account.

If installation is successful, the browser displays the Active Workspace client.

You can also verify the status of Active Workspace Gateway and services using the Active Workspace gateway ping:

**http://myhost:3000/ping**

# 17. Configure heterogeneous operating system environment

If you are adding Windows Teamcenter clients to a Linux Teamcenter environment, you must perform the following tasks:

1. Install Teamcenter and configure the database (Teamcenter application root and data directories) on a Windows system that can serve a common mount point for all Windows clients.

   This allows the Windows and non-Windows Teamcenter clients to interoperate, particularly in volume management.

2. Synchronize the following files in the separate Teamcenter data directories:

   - POM schema files (*TC_DATA***pom_schema_server_sid**)
   - POM transmit files (**\pom_transmit*.sch**)
   - Dataset definition files (*TC_DATA***gs_info*.des**)

3. Make sure your Windows and Linux server configurations contain identical sets of Teamcenter features. For example, if you install features or custom templates on a Linux server, you must install the same features and templates on your Windows server.

4. Configure File Management System (FMS) on Linux and Windows volume servers.

Conversely, if you create a Teamcenter database by running the Teamcenter setup program from a Windows workstation, you must install Teamcenter on Linux clients you want to connect to the database.

# Part IV: Maintain the Teamcenter Environment


Back up your environment after initial installation, and periodically for added security. Add applications and components to Teamcenter environments. Perform database maintenance.

# 18. Back up new installations

Siemens Digital Industries Software strongly recommends backing up new Teamcenter and Oracle installations before using them by performing the following actions:

### Terminate Teamcenter sessions

Prior to upgrade, you must terminate Teamcenter sessions if:

- You are reinstalling or upgrading Teamcenter executables by overwriting an existing Teamcenter data directory. The Teamcenter installation procedure cannot overwrite files when they are in use.
- You are upgrading a Teamcenter database.
- You are migrating an Oracle database to a Windows database server.

1. Instruct all users to check in all Teamcenter business objects, and then close and log off of Teamcenter sessions, including **tcserver** processes.
2. Open a Teamcenter command prompt.
3. Use the **clearlocks** utility to check for nodes connected to the database and remove locks on the database:

   Windows systems:

   ```
   %TC_BIN%\clearlocks -u=Tc-Oracle-user -p=Tc-Oracle-user-password -g=dba
   -assert_all_dead
   ```

   Linux systems:

   ```
   $TC_ROOT/bin/clearlocks -node_names
   ```

4. On Linux systems, note the node names returned, and then type the following command for each node name returned:

   ```
   $TC_ROOT/bin/clearlocks -assert_dead node-name
   ```

   Replace *node-name* with a returned node name.

5. Stop all Teamcenter services, including FMS.

### Back up existing Teamcenter data

If you upgrade a Teamcenter database, back up existing Teamcenter data.

> Caution:
>
> Back up the database, Teamcenter data directory, and all Teamcenter volume directories to an external backup device before performing an upgrade. This provides a safeguard against data loss in case problems occur during the upgrade.

Back up the following directories:

- The Teamcenter application root directory on each installed workstation
- The Teamcenter data directory for each configured database
- The Teamcenter volume directories for each configured database

These are the only directories affected by Teamcenter installation. If you created other directories that contain data used by your existing Teamcenter installation, such as a separate POM transmit schema directory, Siemens Digital Industries Software recommends that you back up these directories as a precautionary measure.

### Back up Teamcenter databases

Back up your Oracle server and databases:

1. **Export existing Oracle databases.**
2. **Terminate Teamcenter-Oracle sessions.**
3. **Back up the Oracle installation.**

# 19. Choose a display language

The default language displayed is the one specified by your operating system locale settings. You can choose to override the default display language if required.

At each logon, you can choose between multiple languages, depending on your company's policy and installation. Choose a language in the logon dialog when you log on to Teamcenter.

Alternatively, you can specify the language in your browser preferences. For example, in Microsoft Edge, choose **Settings→Languages** to add languages or modify language preferences.

Your ability to set the language for the client depends on the character set encoding of the Teamcenter server host and also the character set encoding of the Teamcenter database.

To prevent mixed-language display after you change the client display language, clear your web browser cache. This prevents the interface from displaying in mixed languages.

# 20. Manage environments

## Add or remove software in the repository

Add software to the Deployment Center repository or remove software you no longer need to free space on the software repository machine.

### Add software kits into the repository

1. Download the software kits for the software versions that you want to deploy in your Teamcenter environment.

2. Unzip the software kits and copy the unzipped directories to the *software* subdirectory in one of your registered Deployment Center repository locations.

3. Log on to Deployment Center, and click **SOFTWARE REPOSITORIES**.

   The **Software Repositories** page opens the **Active Media** tab of the repository and displays the **Software Media** table.

4. Check the list of software to verify that the software you added is included in the list. If the list does not update immediately, wait a few minutes for Deployment Center to finish scanning the repository.

   If the software you added still does not appear in the list, verify that you placed the complete, unzipped software kit contents in the **software** directory.

   If necessary, troubleshoot the Deployment Center repository service.

### Remove obsolete software kits from the repository

When a software kit is no longer being used in a registered Teamcenter environment, you can remove it from the **Active Media** list.

1. Open the **Active Media** tab to display all the registered software kits. Click the software kit you want to remove.

2. Click **Remove** ⊖ on the command bar and confirm the deletion.

   If the software is used by an environment, an error message explains that it can't be removed and which environments are using it.

   If the software is not used and is free to be removed, the selected software is moved to the **Obsolete Media** tab. Deployment Center deletes the software and the software directory from the file system.

To complete the software kit removal from Deployment Center, click the **Obsolete Media** tab, select the software to remove, and then click **Remove** on the command bar.

The repository scanner removes the software kit registration and removes the software from Deployment Center database. The selected software no longer appears in the **Obsolete Media** tab.

# Creating environments

## Create an environment in Deployment Center

You can create an environment for your planned deployment. When you are ready to add software to your new environment, Deployment Center displays only the versions of **Available Software** that are supported in a new environment.

### Create an environment

1. Log on to Deployment Center, and click **ENVIRONMENTS**.

   The **Environments** page lists currently planned and registered environments.

2. On the far right below the command bar, click **Add Environment** .

3. The new environment appears highlighted in the list. Choose **Overview** to display its information.

4. You can edit some of the properties, such as **Name** and **Type**. On the command bar:

   Click **Start Edit** to edit properties. To save your changes, click **Save Edits** .

   To cancel your changes, click **Cancel Edits** .

You can also choose to export the configuration of an existing environment. You can reuse its configuration to create another environment using the quick deployment procedure.

## Register an environment in Deployment Center

If you created an environment using Teamcenter Environment Manager (TEM), you can import it into in Deployment Center using the following registration and validation process:

1. **Register the environment using the registration utility.**

2. **Validate and generate an environment report.**

3. **Review the report and perform corrective actions.**

Note:

- The registration utility (**send_configuration_to_dc**) scans only configurations from TEM. Components supported by Deployment Center that were installed by other installers, such as Web Application Manager (insweb), must be added manually after scanning the environment.
- For information about features supported in the current software, generate a software configuration report.
- If your environment contains business logic servers in addition to the corporate server, make sure you scan and register the corporate server first to maintain environment consistency.

## Register an environment

Register an existing environment in Deployment Center by running the command line **send_configuration_to_dc** utility on machines that have TEM-installed Teamcenter software. If the environment is distributed across multiple machines, run the utility on each machine that contains a configuration you created using TEM. Run the utility on your corporate server machine first.

The utility sends TEM configuration information for installed features to Deployment Center.

1. On the Deployment Center machine, locate the registration utility package:

   **Windows systems:**

   *DC-installation***webserver\additional_tools\send_configuration_to_dc.zip**

   **Linux systems:**

   *DC-installation*/**webserver/additional_tools/send_configuration_to_dc.zip**

2. On each machine in the environment, prepare to run the registration utility:

   a. Copy the **send_configuration_to_dc.zip** package to a local directory and unzip the package.

   b. Install a **certified Java runtime environment (JRE)**.

   c. Open an administrator command prompt.

   d. Change to the **send_configuration_to_dc** directory in the path where you unzipped the registration utility.

   e. Set the **JAVA_HOME** and **JRE_HOME** environment variables to the location of the JRE.

   f. Set the **TC_ROOT** environment variable to the Teamcenter installation directory on the given host.

3. Beginning with the corporate server machine, type the **send_configuration_to_dc.bat** command (on Windows systems) or the **send_configuration_to_dc.sh** command (on Linux systems) with the following arguments to register the local Teamcenter configuration from TEM into Deployment Center:

   **-dcurl** (required)

   Specifies the URL for the Deployment Center server.

   **-dcusername** (required)

   Specifies the user name for the Deployment Center administrator.

   **-dcpassword** *or* **-dcpasswordfile** (required)

   Specifies the password for the Deployment Center administrator. You can specify the password as text or use an encrypted password or password file. If the password file path contains spaces, enclose it in quotes.

   **-environment** (required)

   Specifies a name to identify the imported environment in Deployment Center.

   **-config** (optional)

   Specifies the ID of the Teamcenter configuration you want to scan. Specify this argument if you installed multiple configurations in a single **TC_ROOT** location.

   **-machine** (optional)

   Specifies the hostname of the machine. This is not needed in most cases, as the utility reads the machine name from the **configuration.xml** file of the existing installation.

   This argument should be used only in cases where the machine's hostname differs from the hostname specified in the **configuration.xml** file. For example, if you register a system configured with a fully qualified domain name (FQDN) into Deployment Center, include the **-machine** argument.

Example:

```
send_configuration_to_dc.bat -dcusername=dcadmin
   -dcpasswordfile="E:\admin passwords\dcadmin.pwf"
   -dcurl=http://dc_host:8080/deploymentcenter
   -environment=tc_scanned
```

After the scan completes, the utility displays the message:

```
Environment has been generated successfully for review.
```

4. Type the **send_configuration_to_dc.bat** command on the remaining machines to complete scanning of the environment into Deployment Center.

5. After you scan all machines in the environment, review and validate the environment in Deployment Center.

### Generate an environment validation report

1. Log on to Deployment Center.

2. In the **Environments** page, select the name of the scanned environment in the environments list. You specified this name in the **-environment** argument when you ran the **send_configuration_to_dc** utility.

3. In the **Software** tab, note the status of the selected software is **In Review** .

   If problems were encountered when scanning the environment, Deployment Center displays a warning icon above the **Selected Software** list.

4. In the command bar on the far right, click **Validate and Generate Environment Report**:


   When prompted, click **OK** to confirm this action.

Deployment Center generates an *environment validation report* that contains feature mappings, information about scanned TEM environments, and actions to perform to complete the configuration of the scanned environment. The report is placed in the following location:

**Windows systems:**

*Deployment Center-repository***report\EnvironmentValidationReport_***environment-name**.**html**

**Linux systems:**

*Deployment Center-repository*/**report/EnvironmentValidationReport_***environment-name*.**html**

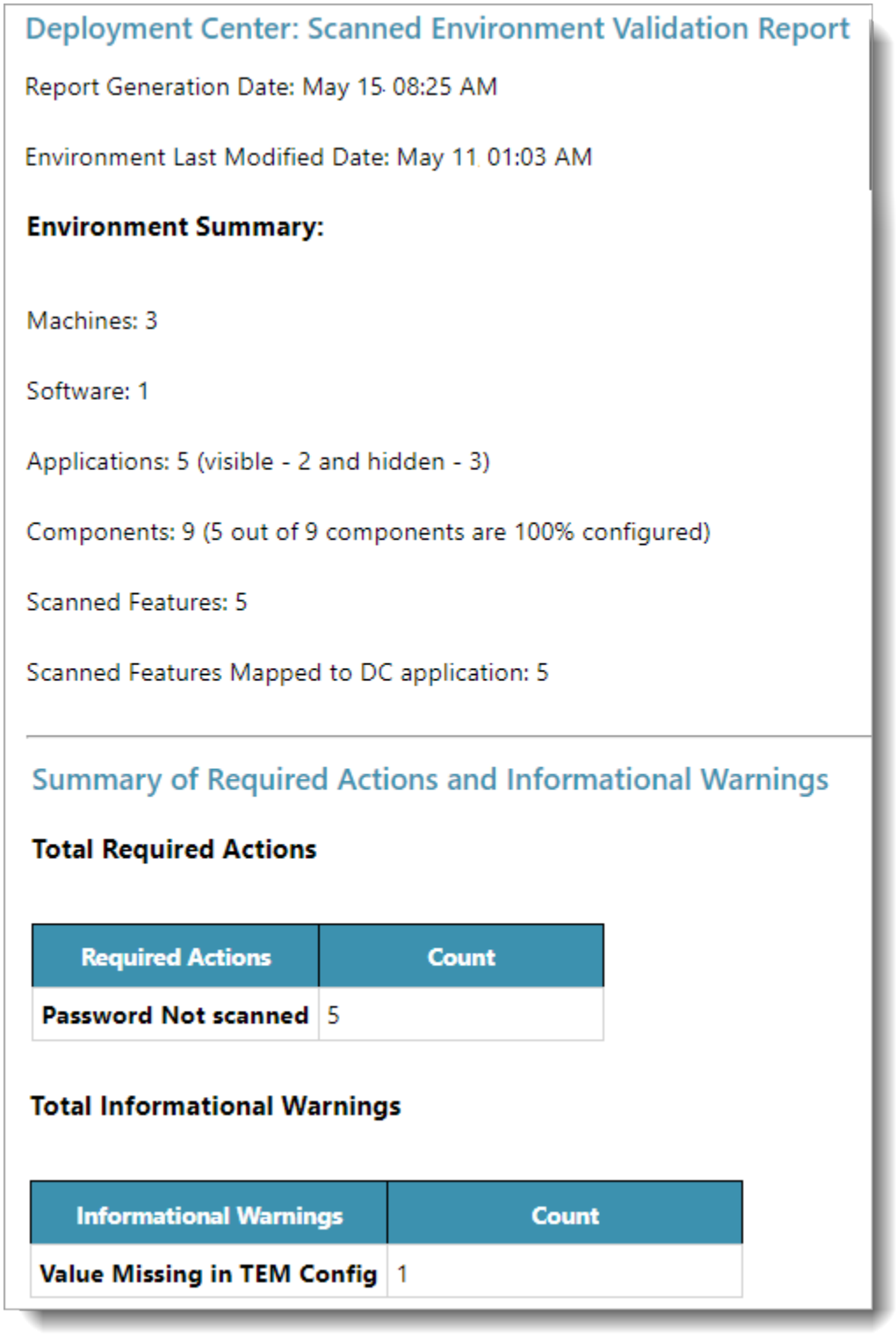
Deployment Center automatically downloads the report, so you can open it through the web browser **Downloads** feature or from the location above.

Alternatively, you can generate a validation report from the command line by running the **send_configuration_to_dc** utility with the **-gvr** (generate validation report) argument:

**send_configuration_to_dc.bat -gvr -dcusername=***DC-user* **-dcpasswordfile=***password-file* **-dcurl=***DC-URL* **-environment=***environment-name* **-machine=***machine-name*

### Review and perform actions

Open the environment validation report and review all information about the scanned environment.



## Deployment Center: Scanned Environment Validation Report

Report Generation Date: May 15 08:25 AM

Environment Last Modified Date: May 11 01:03 AM

**Environment Summary:**

Machines: 3

Software: 1

Applications: 5 (visible - 2 and hidden - 3)

Components: 9 (5 out of 9 components are 100% configured)

Scanned Features: 5

Scanned Features Mapped to DC application: 5

### Summary of Required Actions and Informational Warnings

**Total Required Actions**

| Required Actions | Count |
|---|---|
| **Password Not scanned** | 5 |

**Total Informational Warnings**

| Informational Warnings | Count |
|---|---|
| **Value Missing in TEM Config** | 1 |

Begin with the following primary sections:

- **Summary of Required Actions and Informational Warnings**

Describes total numbers and types of actions required to complete the configuration of the scanned environment.

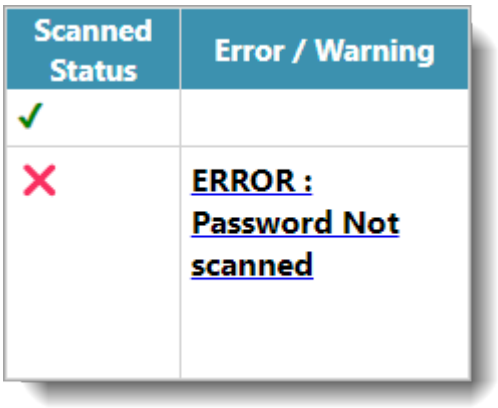
- **Mapping of Scanned TEM Environment Information for This Deployment Center Environment**

  Maps TEM features and GUIDs to Deployment Center applications and components. It maps TEM configuration properties to Deployment Center properties and highlights items that require attention.

Perform recommended actions to complete configuration of the scanned environment:

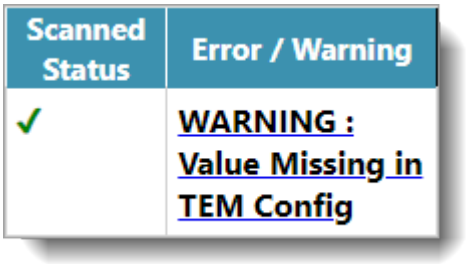
1. In each table in the report, note values in the **Scanned Status** and **Error/Warning** columns.

   In the **Scanned Status** column, a check mark (✔) indicates a valid item, an X () indicates an item that requires corrective action:

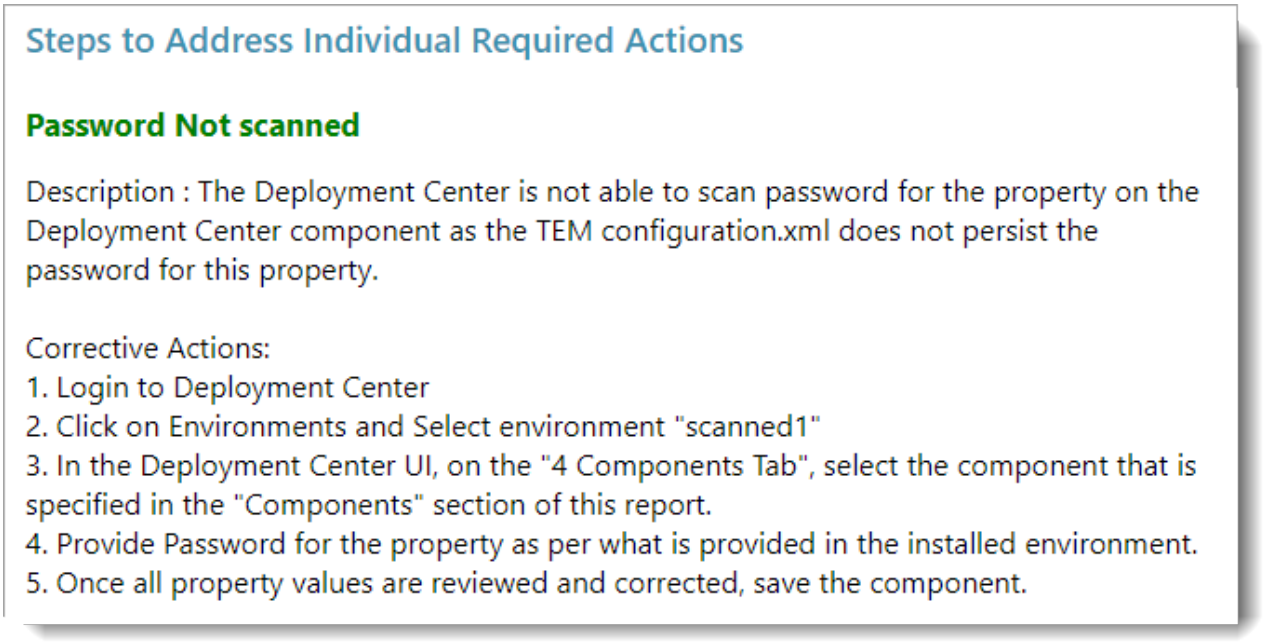



   Note:

   Some items in the report may show a check mark (✔) with a warning. These items are not critical to validation but changes are recommended either during validation or soon after.




2. For each item that error or warning, click the message in the **Error/Warning** column to view the specific required actions, for example:




3. For each issue reported, perform the required actions.

4. When you have performed all actions, generate a new report by clicking **Validate and Generate Environment Report** in the command bar on the far right.

5. If the **Summary of Required Actions and Informational Warnings** section in the new report still lists required actions, perform the required actions and generate a new report again.

6. When the environment validation report prescribes *no* further required actions, the environment is ready to be marked complete. Click **Complete Registration** in the command bar on the far right:


In the **Software** tab, note the status of the selected software is changed to **Installed**  .

Alternatively, you can complete the environment registration from the command line by running the **send_configuration_to_dc** utility with the **-rc** (complete registration) argument:

**send_configuration_to_dc.bat -rc -dcusername=***DC-user* **-dcpasswordfile=***password-file* **-dcurl=***DC-URL* **-environment=***environment-name* **-machine=***machine-name*

To view properties of machines in your scanned environment, go to the Deployment Center home page and click the **MACHINES** tile. From the resulting page, you can view all machines used in deployed Teamcenter environments.

Caution:

If you use TEM to update a configuration that has been scanned as part of an environment in Deployment Center, make sure that you run the **send_configuration_to_dc** utility to update the environment information. Otherwise, configuration changes performed locally on Teamcenter servers since the last time the **send_configuration_to_dc** script ran could be overwritten.

### Importing Java EE web applications

Web Application Manager (insweb) is a separate tool from TEM, and the registration utility (**send_configuration_to_dc**) can only process TEM configuration files. As a result, the **Teamcenter Web Tier (Java EE)** component is not included in a scanned environment. After you scan your environment from TEM, you must add the Java EE web tier component to the environment.

If you use Security Services, you must also add the **Teamcenter Security Services (TcSS)** component.

1. Log on to Deployment Center and go to the **Environments** page. Select the environment you scanned from the list.

2. In the **Components** tab, click **Add component to your environment** ⊕. Add the **Teamcenter Web Tier (Java EE)** component to the environment. And, if you use Security Services, add the **Teamcenter Security Services (TcSS)** component.

3. In the **Selected Components** list, select **Teamcenter Web Tier (Java EE)** and enter the configuration parameters for the web tier from the original environment. If you are unsure of these settings, you can find them using either of these methods:

   - In the Web Application Manager, select your web application and click **Modify**. Then review the web application information in **Modify Web Application**.

   - Review the *.dat* files in the staging location for your web application. For example, find *WEB_ROOT\staging1* on the machine where you run the Web Application Manager.

   Repeat this step for the **Teamcenter Security Services (TcSS)** component, if applicable.

4. Save your settings.

5. Review the remaining **Selected Components** to make sure they are all 100% configured.

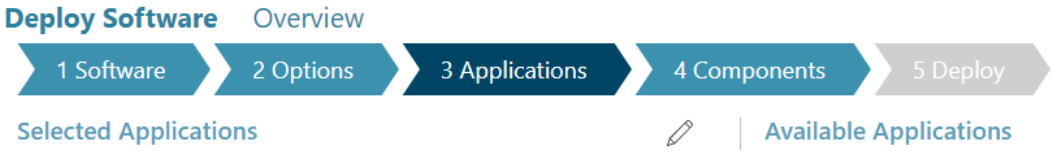
If you experience other problems in registering environments with Deployment Center, see *Deployment Center — Usage*..

# Adding applications and components

## Add applications

*Applications* contain administration data, software modules, and parameters that add specialized functionality to the Teamcenter environment.

### Adding applications using Deployment Center




Select the **Applications** tab to choose applications. The list of available applications is determined by the software you selected in the **Software** tab. Some applications are automatically selected based on your **Selected Software**. For example, if you choose Active Workspace, the **Selected Applications** list includes applications that are required for an Active Workspace installation.

1. In Deployment Center, select your existing environment.

2. In the **Applications** tab, click **Add or Remove Selected Applications**.

   The **Available Applications** panel displays the available applications.

3. In **Available Applications**, choose the applications to install. If an application has dependent applications, Deployment Center automatically selects those additional applications.

4. Click **Update Selected Applications** to add them to the **Selected Applications** list.

   The added applications show **Pending Install** status in the **Selected Applications** list.

   To remove an application that is not yet installed, deselect the application in the **Available Applications** list, and then click **Update Selected Applications**.

5. When your **Selected Applications** list is complete, go to the **Components** tab.

6. In the **Components** tab, note any components whose configuration status is not **100%**. These are either dependent components for your selected applications or components with parameters added by your selected applications.

   For each component, enter required parameter values, and then click **Save Component Settings**.

   Enter required parameter values until all components in the environment show a configuration status of **100%**.

7. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

8. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

### Can I remove an application after it is installed?

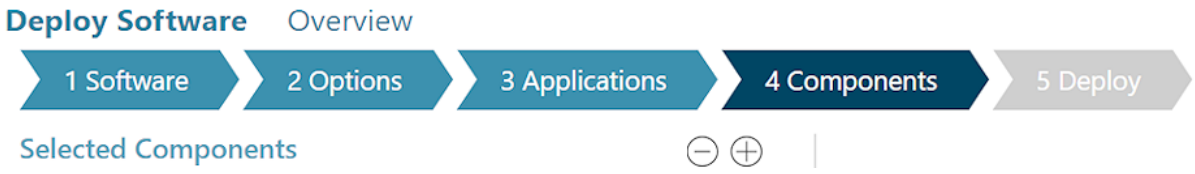
In Deployment Center, removing an installed application is not supported.

> Caution:
>
> If you installed an application using Deployment Center, *do not* attempt to uninstall it using TEM. Deployment Center does not populate the uninstall information required by TEM. Uninstalling an application using TEM may not remove it in Deployment Center, or may cause inconsistencies in the environment in Deployment Center.

## Add components

*Components* are the architectural pieces of Teamcenter, such as servers, services, and databases.

### Adding components using Deployment Center




You select components to install in the **Components** tab in Deployment Center.

Some components are automatically selected based on your selections in the **Software** and **Applications** tab. The list of components available for installation is also determined by your selections in the **Software** and **Applications** tabs. For example, some components require a corresponding application to be selected before the component is made available. Some components are allowed only a single instance within an environment, so if a component is already installed, it may not be in the list of available components.

Configuration parameters for some components may require server names, user names, passwords, URLs, and other system information you may have previously entered for other components in your environment. When you add components, some parameters may be prepopulated with those values from other components. Some prepopulated values may not be editable. For example, in a single box environment, **Machine Name** and **OS** may not be editable.[1]

Some parameters may provide dropdown lists of values from which you can choose. For example, in a distributed environment, the **Machine Name** field for a component may provide a selection list of machine names already defined in your environment.

1. In Deployment Center, select your existing environment.

2. In the **Components** tab, click **Add component to your environment** ⊕ to add components.

   The **Available Components** panel displays the available optional components.

3. In **Available Components**, select the components to install. Then click **Update Selected Components** to add them to the **Selected Components** list.

   In **Selected Components**, the **COMPLETE** column displays the configuration status for each component. If all required parameters are entered for a component, its completion status is **100%**.

4. Click a component in the list to display its parameters in the right panel. This panel initially displays only required parameters. You must enter values for settings that appear in required parameters view. You can toggle the view between required parameters and all parameters:

   **Show all parameters**

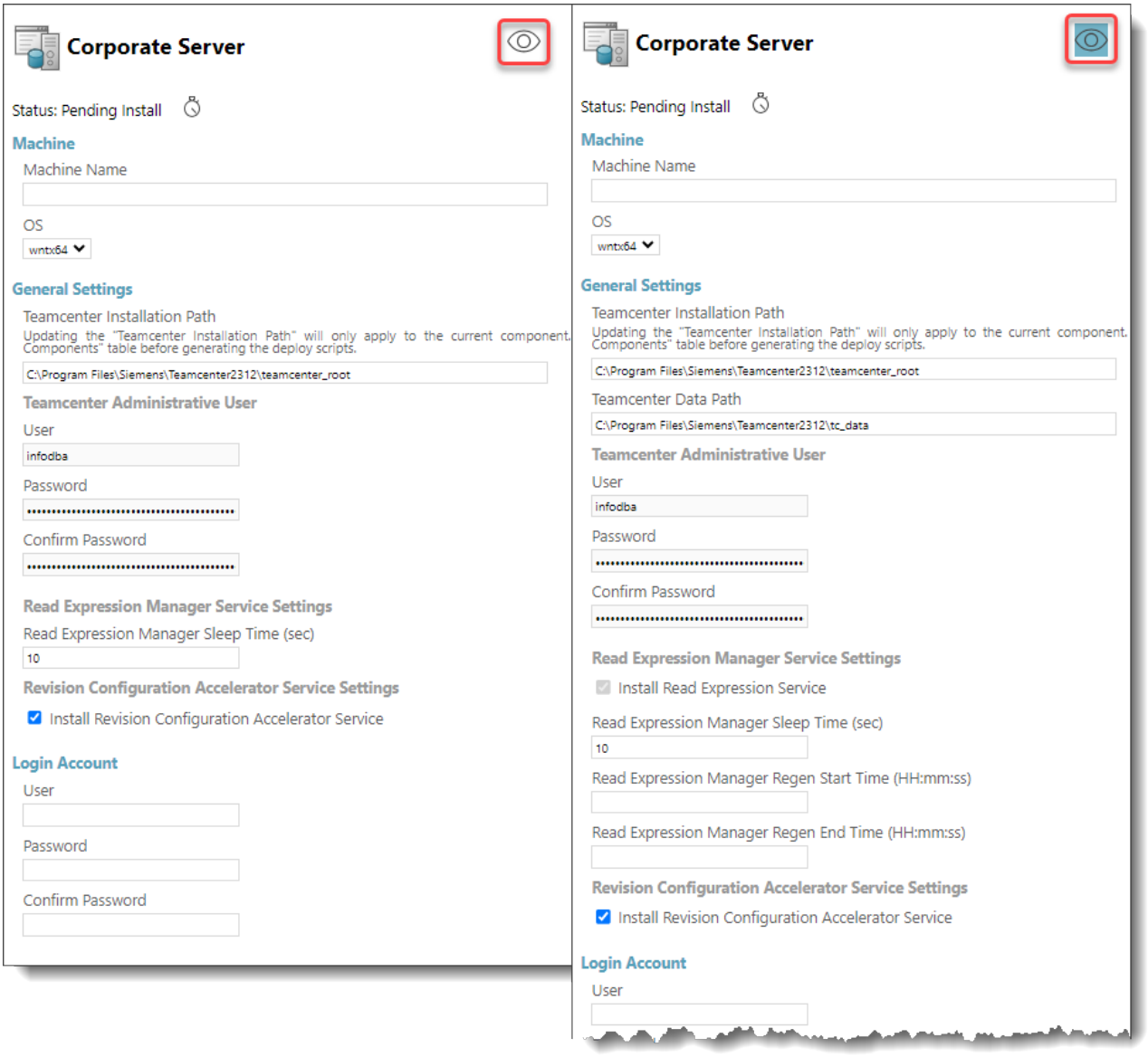
   Required parameters view displays only required parameter information. Click to expand the view to display both required and optional parameters.

   **Show only required parameters**

   All parameters view displays both required and optional parameter information. Click to collapse the view to required parameters.

1 If you selected the **Single Box** environment type in the **Options** tab, all Teamcenter components must reside on the same machine.




5. For each component, enter required parameter values, and then click **Save Component Settings**.

   If you don't have values for all required parameters, you can save your settings at any time and return to finish them. However, the **Deploy** tab is disabled until all components in the environment show a configuration status of **100%**.

6. When all components are fully configured, go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

7. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

If you want to remove a component, you can do so, provided that the component is optional and you have not generated deployment scripts that include the component.

To remove a component from the **Selected Components** list:

1. Click the component you want to remove.

2. From the command bar, click **Remove**⊖. (This option is displayed only for components that are eligible for removal.)

   Deployment Center prompts you to confirm deletion of the component and its dependent components.

   Dependent components that were added to the environment with the main component are also removed for the same machine. Other components of the same type are not removed. For example, if you have two server pools, removing one server pool removes its dependents but the other server pool remains.

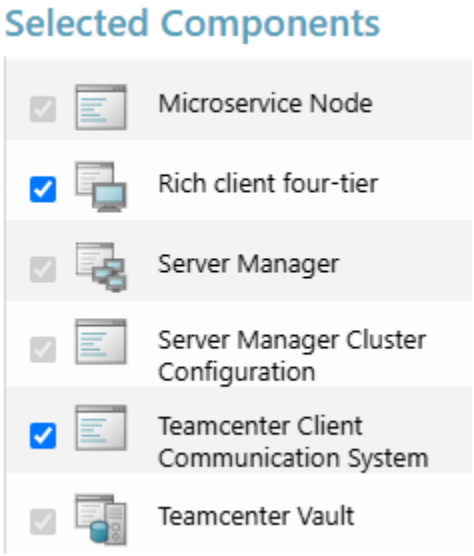
## Uninstall components

### What components can be uninstalled?

Some components can be uninstalled from a deployed Teamcenter environment using the **Uninstall Components** ⊖ option in the **Components** tab in Deployment Center. Components that are supported for uninstallation and that have multiple instances in your environment may be enabled for uninstallation. Uninstalling components enables you to optimize your distribution of components by removing and moving them to other machines.

To identify components in your environment that are eligible to uninstall, click **Uninstall Components** ⊖.

Deployment Center displays the **Component Uninstall Panel**. Components that can be uninstalled are indicated with enabled check boxes ☑ in the **Selected Components** list. Components that cannot be uninstalled are indicated with disabled check boxes ☑:

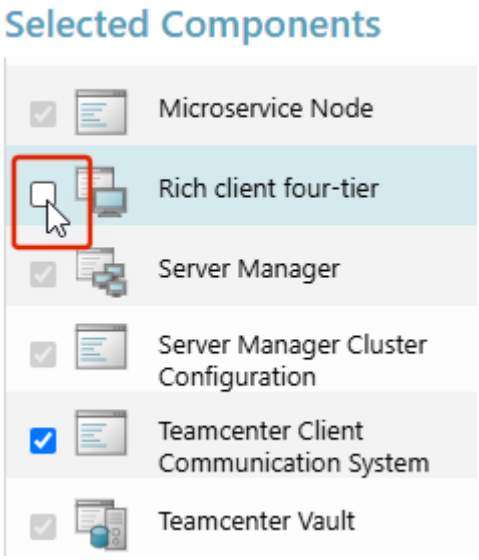



For example, if an environment has two master FSC components, only one can be uninstalled because a Teamcenter environment is required to have at least one master FSC.

### Uninstalling components

When the **Component Uninstall Panel** is displayed, the **Selected Components** list shows all components eligible for uninstallation.

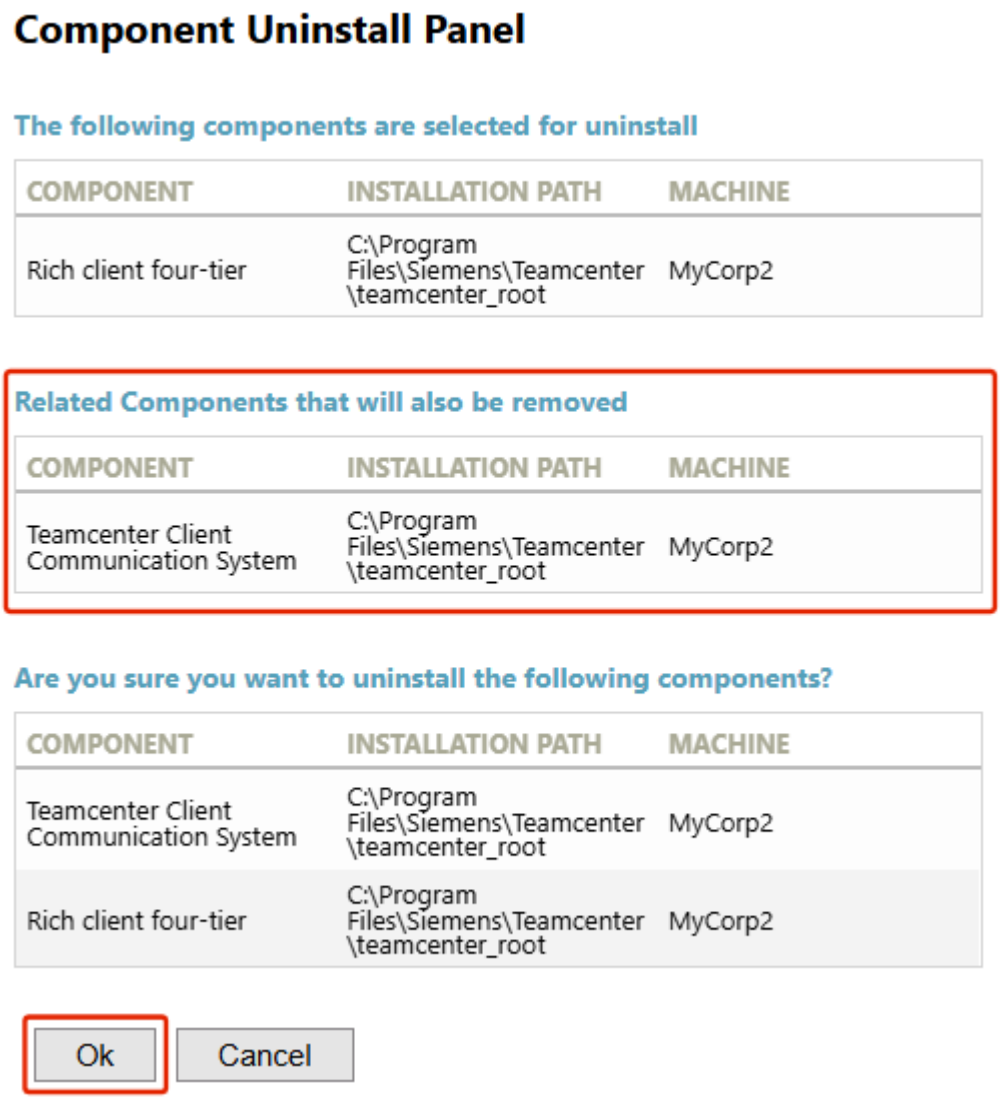
1. To uninstall a component, *clear* its check box ▢:




2. When you complete your selections and deselections for uninstall, click **Uninstall Components**.

   The **Component Uninstall Panel** prompts to you confirm the uninstallation.

If a component to be uninstalled has related components with no dependencies on other installed components, those components are also designated for uninstallation:

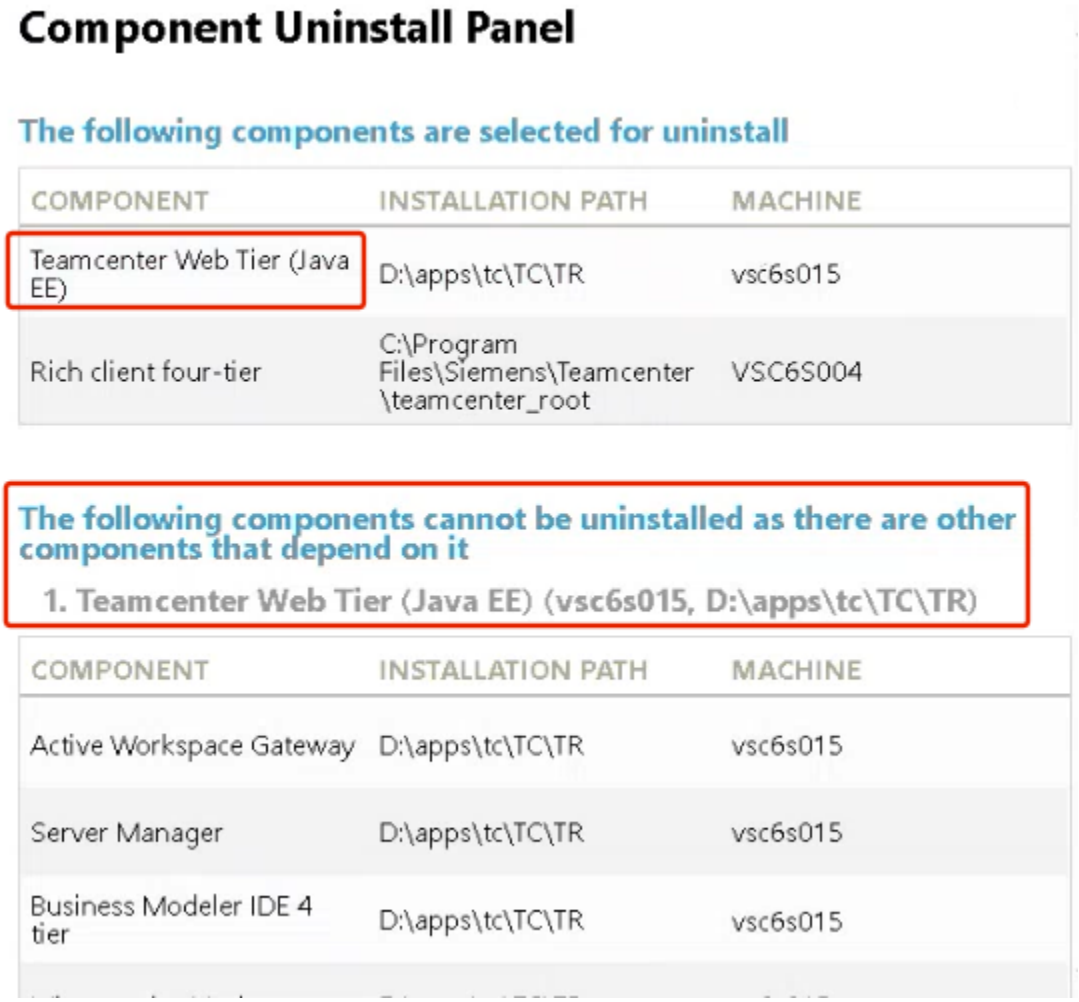



3. Click **OK** to confirm the uninstallation. Components to be uninstalled are indicated with **Pending Uninstall** status.

4. Generate and run deploy scripts to complete the uninstallation on the affected machine.

### Components with dependencies on other components

If a component you designate for uninstallation has dependencies from components *not* eligible for uninstallation, the **Component Uninstall Panel** disallows uninstallation of that component. The **Component Uninstall Panel** displays the dependencies.



Component Uninstall Panel

The following components are selected for uninstall

| COMPONENT | INSTALLATION PATH | MACHINE |
|---|---|---|
| Teamcenter Web Tier (Java EE) | D:\apps\tc\TC\TR | vsc6s015 |
| Rich client four-tier | C:\Program Files\Siemens\Teamcenter \teamcenter_root | VSC6S004 |

The following components cannot be uninstalled as there are other components that depend on it

1. Teamcenter Web Tier (Java EE) (vsc6s015, D:\apps\tc\TC\TR)

| COMPONENT | INSTALLATION PATH | MACHINE |
|---|---|---|
| Active Workspace Gateway | D:\apps\tc\TC\TR | vsc6s015 |
| Server Manager | D:\apps\tc\TC\TR | vsc6s015 |
| Business Modeler IDE 4 tier | D:\apps\tc\TC\TR | vsc6s015 |

### Moving or replacing a component

If you want to replace a component by replacing it with a new instance with different properties, add the new instance before uninstalling the old instance.

Similarly, if you want to move a component to a different machine, install the new instance of the component on that new machine before you uninstall the instance on the old machine.

### Uninstalling components using Quick Deploy

You can uninstall components by adding a **removeQuickDeployComponents** instruction to a quick deploy file.

1. Use the Deployment Center interface to **determine what components can be uninstalled from your environment**.

2. Export your environment to an XML configuration file using either the **Export Environment** option in the Deployment Center interface or the **dc_quick_deploy.bat|sh** utility in export mode.

3. Edit the XML configuration file:

   a. Remove all lines between the **<quickDeployComponents>** and **</quickDeployComponents>** tags.

   b. Insert a **removeQuickDeployComponents** instruction for the component you want to uninstall:

      **<removeQuickDeployComponents>**
      **<removeComponent id="***component-id***" machineName="***machine-name***"**
      **installDirectory="***install-dir***"/>**
      **</removeQuickDeployComponents>**

      Example:

      ```
      <removeQuickDeployComponents>
          <removeComponent id="fnd0_tccs" machineName="myCorp1"
          installDirectory="C:\Program
      Files\Siemens\Teamcenter\teamcenter_root"/>
      </removeQuickDeployComponents>
      ```

4. To perform an optional uninstall impact analysis *without* performing any uninstallation, enter the **dc_quick_deploy.bat|sh** command *without* the **-confirmUninstall** option. The resulting report lists components to be uninstalled and dependencies that may prevent uninstallation.

   **dc_quick_deploy.bat -dcurl=***dc-url* **-environment=***env-name* **-inputFile=***input-file*
   **-dcusername=***user* **-dcpassword=***password*

   Example:

   ```
   dc_quick_deploy.bat -dcurl=http://myCorp:8080/deploymentcenter
   -environment=test_env
     -inputFile=Env_001_quick_deploy_configuration.xml
   -dcusername=dcadmin
     -dcpassword=dcadmin
   ```

5. To perform the uninstallation, run the **dc_quick_deploy.bat|sh** utility with the **-confirmUninstall** option:

   **dc_quick_deploy.bat -dcurl=***dc-url* **-environment=***env-name* **-inputFile=***input-file*
   **-dcusername=***user* **-dcpassword=***password* **-confirmUninstall**

Example:

```
dc_quick_deploy.bat -dcurl=http://myCorp:8080/deploymentcenter
-environment=test_env
  -inputFile=Env_001_quick_deploy_configuration.xml
-dcusername=dcadmin
  -dcpassword=dcadmin -confirmUninstall
```

This command generates deploy scripts to uninstall the specified components from your environment.

Note:

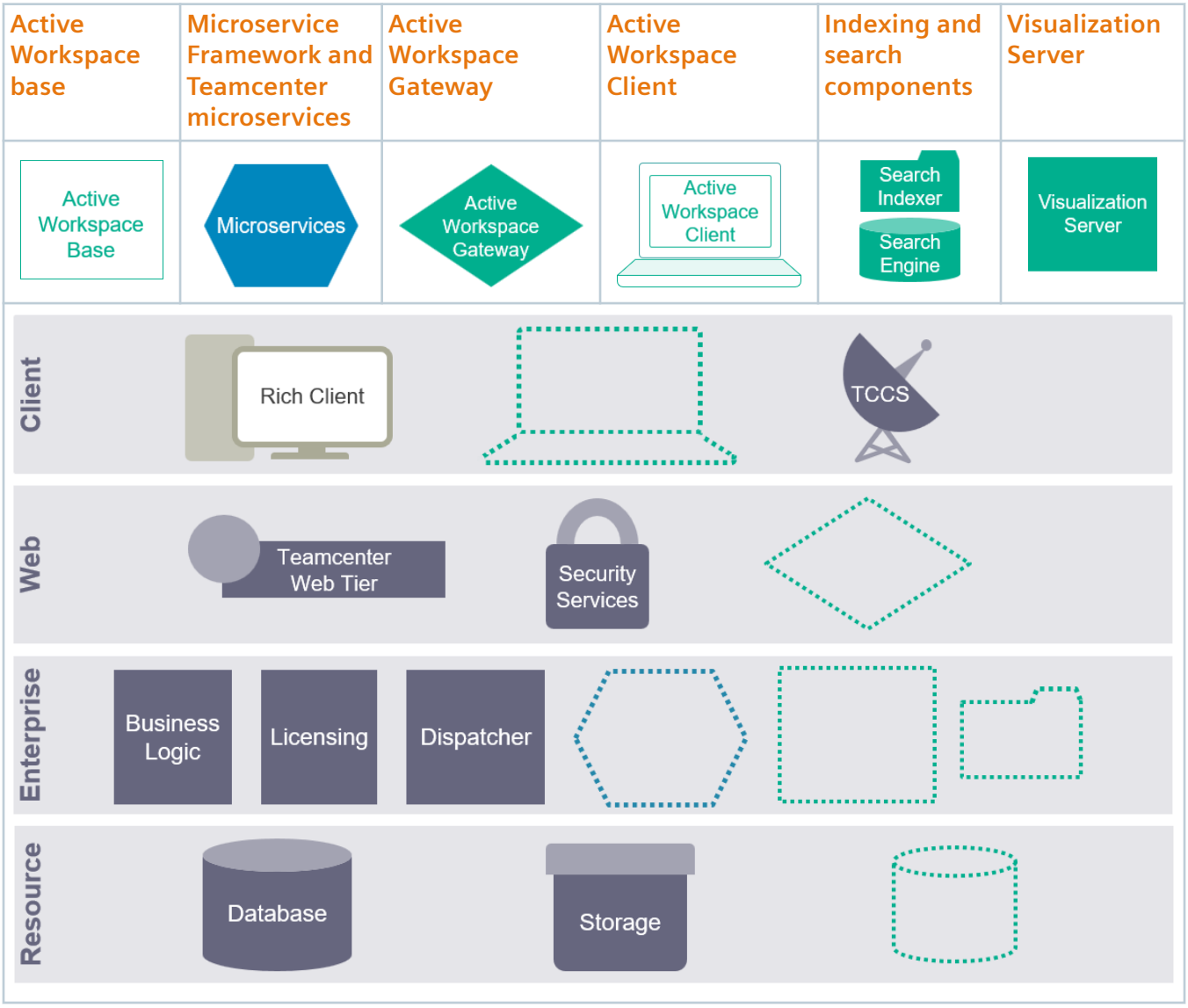
If a component specified for uninstallation is in **Pending Install** status, it is removed from the environment, but no further uninstallation is necessary because the component was not previously installed.

If a component specified for uninstallation is in **Installed** status, its status is changed to **Pending Uninstall**. This status cannot be reverted.

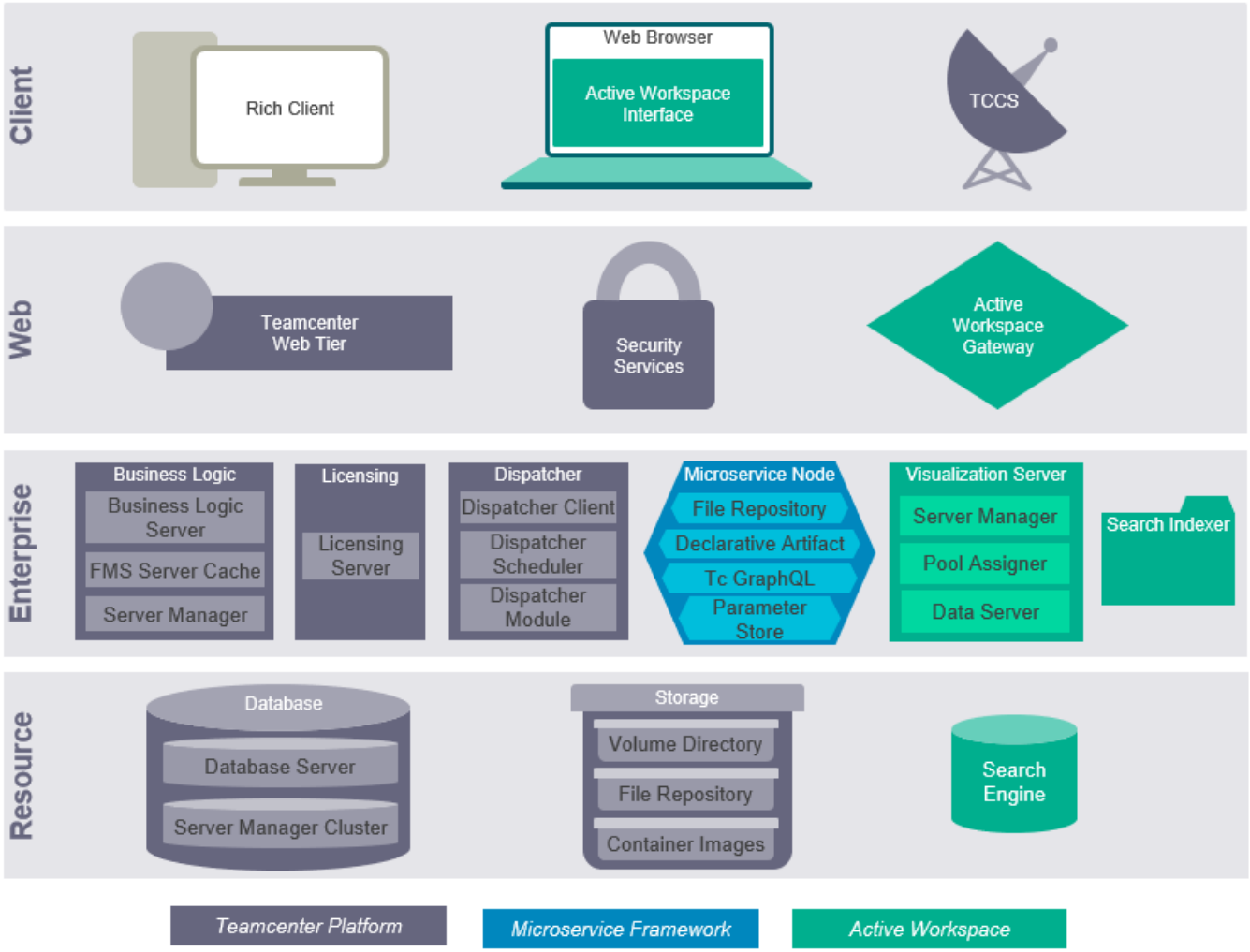
6. Run deploy scripts to complete uninstallation of components.

## Adding Active Workspace to a Teamcenter environment

If you have an existing Teamcenter environment without Active Workspace components, you can add Active Workspace by installing the following Active Workspace and Microservice Framework components in your environment:




After you install components, **deploy the Teamcenter environment** to complete the installation of Active Workspace in your environment.




## Install Active Workspace base

The **Active Workspace Base** application adds components and parameters that add core Active Workspace functionality to your environment.

1. In Deployment Center, select the environment to which you want to add Active Workspace.

2. In the **Applications** tab, click **Add or Remove Selected Applications**.

   The **Available Applications** panel displays the available applications.

3. In **Available Applications**, choose **Active Workspace Base**, and then click **Update Selected Applications**.

   Deployment Center selects additional dependent applications. In a default Teamcenter environment with Active Workspace, these would include:

| | |
|---|---|
| **Client Configuration** | **Reporting** |
| **Document Management** | **Subscription** |
| **Extensions>Active Workspace User Management** | **XRT Editor** |

Optionally, you can select these applications to add them to your Active Workspace deployment.

4. When your **Selected Applications** list is complete, go to the **Components** tab.

5. In the **Components** tab, note any components whose configuration status is not **100%**. These are either dependent components for your selected applications or components with parameters added by the selected applications.

   For each component, enter required parameter values, and then click **Save Component Settings**.

   Enter required parameter values until all components in the environment show a configuration status of **100%**.

6. Go to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines. When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

7. Locate deployment scripts, copy each script to its target machine, and run each script on its target machine.

   For more information about running deployment scripts, see the *Deployment Center — Usage*.

### Install Active Workspace client

Deployment Center adds Active Workspace client components and parameters when you select the **Active Workspace Base application**. Make sure all Active Workspace client components are configured to **100%**, and then generate and deploy scripts as usual.

## Migrate Teamcenter to a different JRE

Change the Java runtime environment (JRE) used by Teamcenter deploy scripts on a given machine.

### Procedure

Perform these steps if you install a different JRE than you used when you initially deployed Teamcenter on the given machine. These steps update the JRE referenced by deploy scripts on the machine.

1. On the machine on which you want to update the designated JRE for Teamcenter, set the following system environment variables to the path to the new JRE:

   - **JAVA_HOME**

- **JDK_HOME**
- **JRE_HOME**
- **JRE64_HOME**

2. Locate the deploy script for the given host.

   Deploy scripts are in the location specified by the **scriptsDir** argument during Deployment Center installation. The default location is *TC_ROOT***dc\deploy_scripts** (on Windows systems) or *TC_ROOT*/**dc/deploy_scripts** (on Linux systems).

3. Run the deploy script on the target machine using the **migrateJRE** option. For example:

   **deploy.bat|sh -dcusername=***DC-user* **-dcpassword=***DC-password* **-migrateJRE**

   Deployment Center updates scripts and properties on the machine to use the JRE specified by the environment variables set in step 1, and restarts affected services on the machine.

   Note:

   Running **deploy.bat|sh** with the **-migrateJRE** option does not perform the deploy actions specified in the deploy script, it *only* migrates settings on the machine to use the specified JRE. If you want to perform deploy actions using the same deploy script, run the script without the **-migrateJRE** option.

# 21. Manage databases

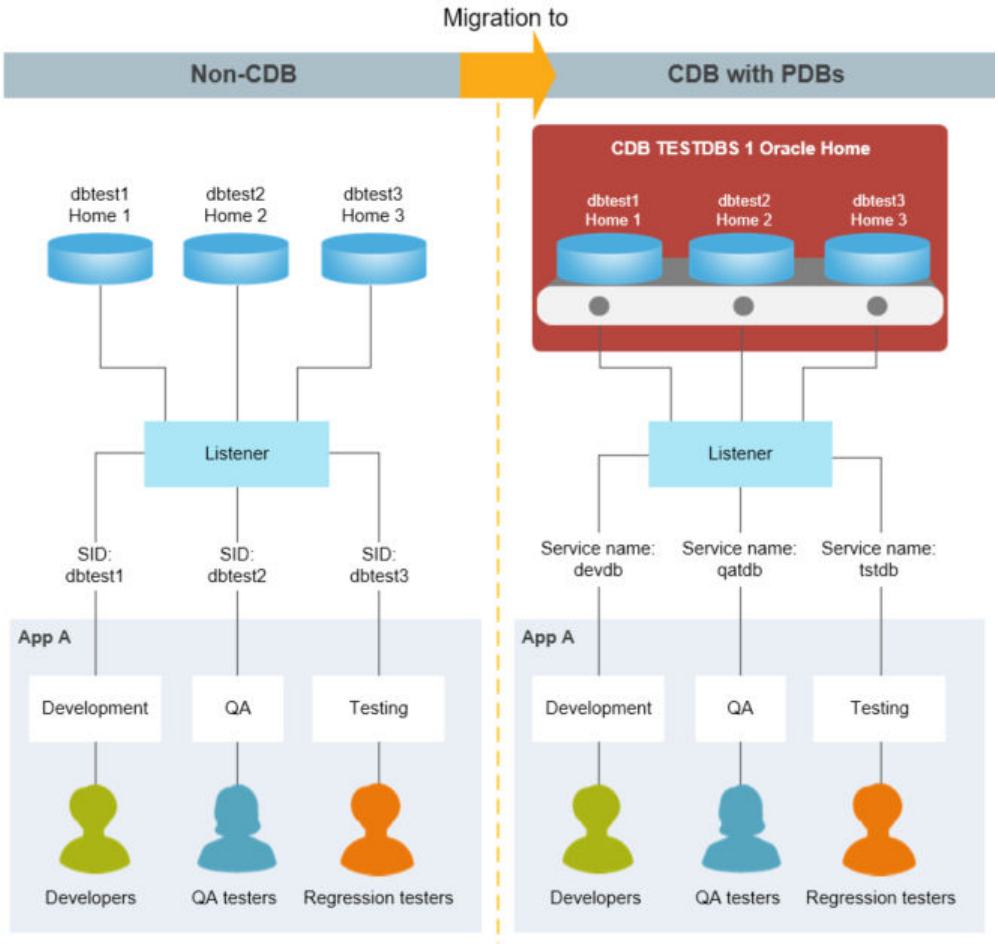
## Migrate a non-CDB database to a CDB database

Teamcenter supports Oracle's **multitenant database architecture** if you use Oracle 12c or later. A multitenant architecture is deployed as a Container Database (CDB) with one or more Pluggable Databases (PDB).

A *Container Database* (CDB) is similar to a conventional (non-CDB) Oracle database, with familiar concepts like control files, data files, undo, temp files, redo logs, and so on. It also houses the data dictionary for objects owned by the root container and those that are visible to databases in the container.

A *Pluggable Database* (PDB) contains information specific to the database itself, relying on the container database for its control files, redo logs and so on. The PDB contains data files and temp files for its own objects, plus its own data dictionary that contains information about objects specific to the PDB. From Oracle 12.2 onward a PDB can and should have a local undo tablespace.

You can **migrate a non-CDB database to a CDB database** using Oracle tools. The following example illustrates the database architectures before and after migration.




Teamcenter supports CDB and non-CDB databases. Be aware that **Oracle has deprecated support for non-CDB databases** and may discontinue support after Oracle 19c.

If you migrate a non-CDB Teamcenter database to a CDB database, you must perform the migration *after* you upgrade to Teamcenter 2412.

## Change the Oracle password

If you use an Oracle database and want to change the password Teamcenter uses to connect to the database, you can do this two ways using the **install** utility:

- **Encrypt the password file** using the **-encryptpwf** argument.
- **Encrypt the database connection string** using the **-encrypt** argument.

### Encrypt the password file

To encrypt a password file, you set a temporary environment variable to the password you want to encrypt, and then generate an encrypted password file using the **-encryptpwf** argument for the **install** utility.

1. Open a Teamcenter command prompt.

2. Create a temporary environment variable and set it to the password you want to encrypt:

   ```
   set variable-name=password
   ```

   For example:

   ```
   set temp_pw=mypassword
   ```

   For security, choose a unique and obscure name for the environment variable, and delete the variable promptly after completing this procedure.

3. Type the following command:

   ```
   install -encryptpwf -e=variable-name -f=password-file
   ```

   Replace *variable-name* with the name of the environment variable you created. Replace *password-file* with the path and name of the password file to create. For example:

   ```
   install -encryptpwf -e=temp_pw -f=pwd.txt
   ```

   This command generates an encrypted password file that can be used for connecting to the Teamcenter database. The password file can also be used with Teamcenter utilities that use the password file (**-pf**) argument.

4. Delete the temporary environment variable you created in step 2.

   > Caution:
   >
   > This step is important for security.

### Encrypt the database connection string

To encrypt the database connection string, you must temporarily set the **TC_DB_CONNECT** environment variable and then re-encrypt the connection string using the **-encrypt** argument for the **install** utility.

1. Open a Teamcenter command prompt.

2. Set the **TC_DB_CONNECT** environment variable:

```
set TC_DB_CONNECT="db-user:password@database-ID"
```

Replace *db-user* with the database user name (the Oracle user). Replace *password* with the new database password. Replace *database-ID* with the Oracle database name.

3. Type the following command:

```
install -encrypt
```

This command generates a new database connection string with the new Oracle password encrypted. Copy the new database connection string.

4. Open the Teamcenter environment variables script for editing:

**Windows systems:** Open the *TC_DATA***tc_profilevars.bat** file in a plain text editor.

**Linux systems:** Open the *TC_DATA*/**tc_profilevars** file in a plain text editor.

5. Locate the following line in the file:

```
set TC_DB_CONNECT=connection-string
```

6. Replace the existing *connection-string* with the string generated by the **install -encrypt** command.

7. Save the changes to the **tc_profilevars** file.

# Part V: Appendices

Supplemental procedures and references for installing Teamcenter and Active Workspace.

# 22. Troubleshooting

## Troubleshooting Teamcenter server installation

### Installation log files

Teamcenter Environment Manager generates files in the **install** directory under the Teamcenter application root directory.

- **install***date-time_configuration-ID*.**log**

  Teamcenter Environment Manager generates a log file for each installation and configuration you create. The log file contains a record of activities performed by Teamcenter Environment Manager. Keep these files to maintain a complete history for troubleshooting purposes.

- **configuration.xml**

  This file contains a record of the Teamcenter installation. Teamcenter Environment Manager uses the configuration file to enable you to maintain the installation, including adding and removing components, patching installations, and upgrading installations.

  > Caution:
  >
  > Do not remove the **configuration.xml** file. Removing the **configuration.xml** file results in the inability to modify the installation using Teamcenter Environment Manager.

- **uninstall.xml**

  This file contains a record of the Teamcenter uninstallation.

In addition, auxiliary programs called by Teamcenter Environment Manager generate files in the **logs** directory under the Teamcenter application root directory. Most files have the format:

*program-name*.**syslog**
*program-name*.**log**

Of these files, the system log (**.syslog**) files usually contain the most relevant error data.

## Problems/error messages

See the following information for help resolving errors encountered during Teamcenter installation.

| Problem/error message | Possible cause | Solution |
|---|---|---|
| TEM does not start, reports JRE not found. | JRE path is not set in the system environment. | Set the **JRE_HOME** or **JRE64_HOME** environment variable to specify the path to the required Java Runtime Environment (JRE).<br>For more information, see *System requirements*. |
| | JRE path is set incorrectly in the system environment. | Make sure the path specified in the **JRE_HOME** or **JRE64_HOME** environment variable is correct.<br>For more information, see *System requirements*. |
| Siemens License Server reports an error similar to the following:<br>`Cannot find license file.` | Make sure the **SPLM_LICENSE_SERVER** system environment variable contains the correct port and host name of the Siemens License Server, for example, **29000@myhost** . | If a path in the **CLASSPATH** environment variable contains whitespace characters, those paths must be enclosed in double quotes (**"**). For example:<br>`"C:\Program Files\Microsoft\Web Platform Installer";D:\TcSE\apache-ant-1.9.4\bin` |
| An error similar to the following is displayed during a Teamcenter installation, upgrade, or patch:<br>`Error: Could not find or load main class files.` | The **CLASSPATH** environment variable contains an incorrectly formatted path. | If a path in the **CLASSPATH** environment variable contains whitespace characters, those paths must be enclosed in double quotes (**"**). For example:<br>`"C:\Program Files\Microsoft\Web Platform Installer";D:\TcSE\apache-ant-1.9.4\bin` |

| Problem/error message | Possible cause | Solution |
| --- | --- | --- |
| Running Teamcenter in an IPv6 network environment, the Teamcenter client does not connect to the server at all or hangs when trying to connect to the server. | Some Teamcenter components are sensitive to link-local IPv6 addresses. You must make sure your hosts have global IPv6 addresses and use those when connecting to the Teamcenter server. Problems can occur if you use local-link IPv6 addresses. | Find your IP address using the **ping** or **nslookup** command. Make sure these commands find the a global IPv6 address, not a link-local IPv6 address.<br><br>If not, or if you are unsure, contact your network administrator. Make sure your host name resolves to a global IPv6 address, not a link-local IPv6 address.<br><br>You can also view your host's network addresses using the **ipconfig** command (on Windows systems) or the **ipconfig** command (on Linux systems). |
| During logon using Kerberos authentication, Teamcenter displays the following error:<br><br>`Mechanism level: Clock skew too great` | The system clock time on the Teamcenter client is significantly different from the system clock time at the Kerberos Key Distribution Center (KDC). | Synchronize the system clock times between the Teamcenter client and the KDC. |
| Database daemon services do not start.<br><br>These can include the following:<br><br>• **Teamcenter Task Monitor Service**<br>• **Teamcenter Subscription Manager Service**<br>• **Teamcenter Action Manager Service**<br>• **Teamcenter Tesselation Manager Service** | If the database daemon services run on the same host as the database server, the database daemons may attempt to start before the database server is fully running. If this happens, the daemons fail to start. | If the database daemons run on the same host as the database server, perform one of the following steps:<br><br>• Manually start the database daemon services after the database server is started.<br><br>• Modify the startup settings for the database daemon services to create a dependency on the database service. This ensures the daemons do not start before the database server is fully running. |
| During an installation or upgrade, the FMS server cache (FSC) reports a startup failure with a message similar to the following:<br><br>`Installation interrupted due to the following reason:` | Another service on the same host was running on the same port that the FSC is configured to use. This causes a fatal error to the FSC and the FSC startup log shows a bind exception on the port.<br><br>Some services, such as JBoss, allow the FSC to bind to its port, resulting in failure of the FSC to start, but no errors in the FSC log. | Change the FSC settings to use a different port. |

| Problem/error message | Possible cause | Solution |
| --- | --- | --- |
| `Processing <upgrade> of feature FMS Server Cache failed: FSC service failed to start with an error 1`<br><br>However, the FSC startup log shows no errors and indicates the FSC is left running. | | |
| `Client credential too weak` | This problem can occur on SUSE Linux 11 when executing the following ODS startup command:<br><br>`/ods -u=`*`Tc-admin-user`*<br>`-p=`*`Tc-admin-pw`*` -g=dba`<br><br>The command shell displays the following error:<br><br>`Cannot register service: RPC: Authentication error; why = Client credential too weak`<br><br>`pid = 31635, unable to register (ODSPROG = 536875585, ODSVERS = 1)`<br><br>This problem can also occur when the **idsminetd** program is used for custom Multi-Site configuration. | Restart the remote procedure call (RPC) portmapper service (**rpcbind**) with the following options:<br><br>`kill -15 `*`rpcbind-process`*<br>`rpcbind -i -w`<br><br>Alternatively, you can set the **rpcbind** startup scripts to always run with the **-i** option. The **/etc/sysconfig/rpcbind** file controls this on SUSE Linux. This may vary on other Linux variants. |

## Update Manager FTP errors

The following table describes errors that can occur while connecting to the update server or while downloading updates.

| Error | Resolution |
| --- | --- |
| Cannot contact server | Host or port may be incorrect. Check **Host** and **Port** values and try again. |
| Cannot log on | User name or password may be incorrect. Check **User** and **Password** values and try again. |
| Incorrect Path | Path to the directory on the update server may be incorrect. Check the path and try again. |
| Timeout Error | The update manager received no response from the update server. Try connecting later or contact your system administrator for assistance. |
| Transfer Error | Contact with the update server was interrupted. Try your operation again or contact your system administrator for assistance. |

## Troubleshooting microservices

| Problem/error message | Possible cause | Solution |
| --- | --- | --- |
| 404 error for a microservice request with the Service Dispatcher logging a message that the HTTP header is too large. | In a deployment with a load balancer configured, due to the addition of large cookies by the load balancer, some requests exceed the limit for the header size. | Create a **CUSTOM_REQUEST_BUFFER_SIZE** environment variable and set its value higher than the default microservice service dispatcher request buffer size of 8192 (8 KB), and then restart the service dispatcher. |

## Troubleshooting four-tier architecture deployment

Identify the problem you encountered in your four-tier rich client architecture and perform the solution described.

| Problem | Solution |
| --- | --- |
| Cleaning FIFO entries in **/tmp/tctp** disables server manager, MUX, and TcServer processes. | On Linux hosts, if the server manager is running when the **/tmp** directory is cleaned up by deleting its entries, Teamcenter Transfer Protocol (TCTP) is disabled. Running TcServers cannot accept new requests. The server manager no longer accepts server ready health notifications, so new servers are not published, and new user sessions will get a "no servers available" error.<br><br>In some customer environments and some operating systems, including Redhat Linux, the **/tmp** directory may be automatically cleaned up periodically at a time other than boot time, |

| Problem | Solution |
| --- | --- |
| | particularly files that have not been used recently. Also, the **/tmp** directory may be mapped to memory, and need to be cleaned up often. See the **tmpwatch** command, which is often run as a cron job.<br><br>To configure the location of the TCTP FIFO entries to a directory not monitored by **tmpwatch**, set the **TC_PIPE_NAME_PREFIX** environment variable to the location of the FIFO entries, to avoid locations that are automatically cleaned. |
| Out-of-memory error during a call to **getAttrMappingsForDatasetType** | If you use WebSphere and this occurs when launching NX from the rich client, you must **modify the JVM arguments** in WebSphere to increase memory allocation. |
| Error messages about the server manager pool ID | These messages indicate that the pool ID is in use by another server manager in the cluster. Either place the server managers in different clusters or configure a distinct pool ID. |
| Configuration is correct, but run-time errors occur | Determine from logs whether users are frequently losing a server due to the server timing out and are then having a new server assigned.<br><br>Server startup can consume a great amount of CPU. Consider increasing timeout values and/or the pool size. |
| CFI_error displays when running AIE export in batch mode | When you run AIE Export in batch mode, Teamcenter displays a CFI error. This error occurs because **jt.exe** (Microsoft Task Scheduler) file is missing from the **%WINDOWS%** directory.<br><br>To resolve this problem, download the Microsoft Task Scheduler from the Microsoft Developer Network:<br><br>**https://msdn.microsoft.com** |
| Chinese characters are displayed as square blocks in the Teamcenter rich client. | If you use a nonnative language operating system version of Windows, you must install and enable the Multilingual User Interface (MUI) pack to ensure the language font is displayed properly.<br><br>1. Download and install the MUI pack for Windows from Microsoft.<br><br>2. Open the **Regional and Language Options** dialog box in the Windows Control Panel.<br><br>3. In the **Languages** tab, set the required language for the menus and dialogs.<br><br>4. In the **Advanced** tab and the **Regional Options** tab, set the required language. |

| Problem | Solution |
| --- | --- |
| Teamcenter web application fails to deploy on JBoss (WildFly) with the following error message:<br><br>`Did not receive a response to the deployment operation within the allowed timeout period [60 seconds]. Check the server configuration file and the server logs to find more about the status of the deployment.` | The Teamcenter web application takes longer than the default 60 seconds the JBoss (WildFly) deployment scanner allows for deployments. Add the **deployment-timeout** attribute to the **deployment-scanner** element and set the value to at least **600** seconds before attempting to deploy the web application.<br><br>`<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">`<br>`<deployment-scanner path="deployments"`<br>`relative-to="jboss.server.base.dir" s`<br>`scan-interval="5000"`<br>**`deployment-timeout="600"`**`/>`<br>`</subsystem>` |
| Long running service request that crosses firewalls or proxy servers results in closed connections. | If a user is performing a time-consuming action such as running a large BOM expansion, the server may not respond for 15 minutes or more. When this happens across a firewall, or other proxies, the firewall might automatically close the perceived idle connection. This results in a closed connection in the client application and loss of data.<br><br>To avoid exceeding these idle connection time limits, enable TCP keepalive functionality in the operating system (OS) of at least one of the machines on the client or server side of the each of the HTTP connections between the client applications and the Teamcenter server.<br><br>For example:<br><br>• If a client machine connects to web tier machine, enable TCP keepalive in the OS of the machine where the web tier server runs. This supports both the HTTP connection between client applications and the web tier, and the HTTP connection between the web tier and the Teamcenter server (Server Manager/MUX).<br><br>• If you use a reverse proxy server between a client machine and the web tier machine, enable TCP keepalive in the OS of the machine where the reverse proxy runs.<br><br>If your network configuration requires you to *not* enable TCP keepalive on the TCP endpoint (such as a proxy server), you must enable keepalive in the OS on each *client* machine.<br><br>On Windows machines, enable TCP keepalive by setting the appropriate Windows registry keys. On Linux machines, set TCP keepalive using kernel parameters. See your operating system documentation for information on how to enable TCP keepalive. |

| Problem | Solution |
| --- | --- |
| | Note:<br>TCP keepalive is enabled in Teamcenter client and web tier software by default, and only requires TCP keepalive in the OS of affected hosts to be enabled.<br>Alternatively, if you do not want to enable TCP keepalive, you can increase the timeout setting in the firewall to allow requests to complete. |

## Troubleshooting the .NET web tier

### Resolving .NET server manager port conflicts

When starting the .NET Server Manager Service, Teamcenter may display a message that no Teamcenter servers are available. This can be caused by a port conflict.

To diagnose and resolve this problem, perform the following steps.

1. Open the following file in the *TC_ROOT***net_servermanager\logs** directory:

   **TcServerManager_***timestamp***.log**

2. Search the log file for errors similar to the following example:

```
2014-02-12 21:06:33 [6] ERROR Teamcenter.Enterprise.ServerManager.ServerPoolManager
[(null)] - Remoting port configured for Pool ID: TcPoolA, is already in use. Stop
and start server manager on a different port. Message is: Only one usage of each
socket address (protocol/network address/port) is normally permitted
```

3. If you find an error that states a remoting port is already in use, another process is using the same port as the .NET server manager.

To resolve this problem, either change the .NET server manager port to different value or stop the other process that uses the .NET server manager port.

You can use the Windows **netstat** utility to view all TCP ports currently in use by the system. For example, typing **netstat -a -b** or **netstat -aon** lists the TCP ports currently in use.

# Troubleshooting Oracle

## Finding Oracle errors

When Oracle detects an error, an error code is displayed in the system console window and written to the Teamcenter trace and log files. To assist troubleshooting, Oracle embeds object names, numbers, and character strings in error messages.

The **oerr** utility provides additional troubleshooting information. Often, the additional information offers a solution to the problem.

### View additional information about an Oracle error message

1. Manually set the Oracle environment by entering the following command:

   ```
   export ORACLE_HOME=/u01/app/oracle/product/oracle-version
   ```

   Replace *oracle-version* with the installed Oracle version, for example, **920**.

2. Enter the following command:

   ```
   $ORACLE_HOME/bin/oerr facility error-number
   ```

   Replace *facility error-number* with the Oracle error code, for example **ORA 7300**. ORA is the facility and 7300 is the error number.

   This command displays cause and action messages that you can use to troubleshoot the problem.

# Troubleshooting Microsoft SQL Server

### Microsoft SQL Server 2014 performance is poor

If you migrate a database application to Microsoft SQL Server 2014 from a previous version, the database server may consume excessive CPU resources and cause poor performance.

To correct this problem, change the SQL Server 2014 Compatibility Level setting from SQL Server 2014 (**120**) to SQL Server 2012 (**110**).

For more information about this issue, see the following Microsoft support article:

**https://msdn.microsoft.com**

### Teamcenter update fails with ODBC error

When upgrading a Microsoft SQL Server server, an error similar to the following can occur:

```
++++++++++++++++++++++++++++++++++
ODBC error. SQLSTATE: 42000 Native error: 5074
Message: [Microsoft][ODBC SQL Server Driver][SQL Server]The column '***'
is
dependent on column '***'.
ODBC error. SQLSTATE: 42000 Native error: 4922
Message: [Microsoft][ODBC SQL Server Driver][SQL Server]ALTER TABLE
ALTER COLUMN
<name> failed because one or more objects access this column.
++++++++++++++++++++++++++++++++++
```

This error occurs when the upgrade process attempts to modify a column that has a dependent column with an index. Microsoft SQL Server does not allow changes to columns with indexes. Also, local DBA indexes may exist that don't match the standard OOTB template for indexes, so it was not anticipated.

This problem can happen because columns that have manually-created statistics attached cannot have their properties modified without first dropping the statistics object. This to ensure the statistics object accurately reflects the content of the column. Manual creation of statistics objects is important to ensuring query performance if you set `AUTO_CREATE_STATISTICS = OFF`.

An auto-created statistics object does not prevent a modify action to a column because auto-created statistics objects can be removed automatically. But, if the system encounters a manually-created statistics object, it cannot be removed automatically, and may result in an access error.

To resolve this problem, perform the following steps:

1. Delete the index ***.
2. Delete the dependent column ***.
3. Continue the upgrade.
4. Run the **index_verifier** utility to re-create standard OOTB indexes:

   ```
   index_verifier -u=tc-admin -p= -g=dba -o=DO_IT
   ```

   Replace *tc-admin* with the Teamcenter administrative user.

## Troubleshooting Lifecycle Visualization

Certain software libraries are required to run Lifecycle Visualization on SUSE Linux platforms. If the required libraries are not installed on your system, Lifecycle Visualization may display an error that contains the following text:

```
error while loading shared libraries
```

If this occurs, you must install the missing required libraries.

To display a list of the required RPM packages for Lifecycle Visualization on SUSE Linux, type the following command:

```
env LD_LIBRARY_PATH=Linux_x86_64_SuSE/bin_64 rpm -qf `ldd
Linux_x86_64_SuSE/bin_64/* |
     & egrep '/lib/|/lib64/' | awk '{print $3}' | sort -u` | sort -u
```

From the resulting output, identify the missing libraries and install them on your system.

## Tuning WebSphere JVM memory consumption

If your Teamcenter application requires more memory than what is currently allocated in WebSphere, out-of-memory errors can occur. For example, if you use the NX Integration and attempt to launch NX from the rich client, Teamcenter may report an out-of-memory error during a call to **getAttrMappingsForDatasetType**.

If errors like this occur, you must modify the JVM arguments in WebSphere to increase memory allocation. For information about how to modify JVM arguments, see the IBM support article titled *Setting generic JVM arguments in WebSphere Application Server* at the following site:

**http://www-01.ibm.com**

Before you tune JVM arguments, use memory profiling tools to analyze your memory issues and determine which tuning options you need to use. The following table provides some suggestions, but these may not be suitable in all cases.

**JVM options for tuning the WebSphere Application Server memory usage**

| JVM option | Description | Typical default value | Suggested value |
|---|---|---|---|
| **-Xms** | Controls the initial size of the Java heap.<br><br>Properly tuning this parameter reduces the overhead of garbage collection, improving server response time and throughput. For some applications, the default setting for this option may be too low, resulting in a high number of minor garbage collections. | 50 MB | 512 MB |
| **-Xmx** | Controls the maximum size of the Java heap.<br><br>In general, increasing the minimum/maximum heap size can improve startup, reduce the number of garbage collection occurrences, and increase the throughput until the heap no longer resides in physical memory. After the heap begins swapping to disk, Java performance suffers drastically. Therefore, The heap sizes should be set to values such that the maximum | 256 MB | 1024 MB |

| JVM option | Description | Typical default value | Suggested value |
|---|---|---|---|
| | amount of memory the VM uses does not exceed the amount of available physical RAM. | | |
| **-XX:PermSize** | Sets the section of the heap reserved for the permanent generation of the reflective data for the JVM. This setting should be increased to optimize the performance of applications that dynamically load and unload many classes.<br><br>**PermSize** memory consumption is in addition to the **-Xmx** value set by the user on the JVM options. Setting this to a value of 128 MB eliminates the overhead of increasing this part of the heap. | Client: 32 MB<br><br>Server: 64 MB | 128 MB |
| **-XX:MaxPermSize** | Allows for the JVM to be able to increase the **PermSize** setting to the amount specified.<br><br>Initially, when a VM is loaded, the **MaxPermSize** is the default value, but the VM does not actually use that amount until it is needed. If you set *both* **PermSize** and **MaxPermSize** to 256 MB, the overall heap increases by 256 MB in addition to the **-Xmx** setting.<br><br>If an application needs to load or reload a large number of classes, the following error may result:<br><br>`messageOutOfMemoryError: PermGen space`<br><br>Typically, this means that the JVM started with an insufficient maximum value for permanent generation. | N/A | 256 MB |

## Troubleshooting document rendering

If you are not successful rendering document revisions to translate dataset files, your administrator should review your installation and configuration systematically and verify the following requirements are met.

- Installation of Teamcenter lifecycle visualization Convert software is required by the **previewservice** feature.
  - You must select the **Convert** feature; the **Print** feature is optional.
  - The destination installation directory name must not contain spaces.

- To accommodate high levels of input and output, modify the **vvcp.ini** file on Windows systems, or the **vvcp.***platform***.cfg** file on Linux systems.

  ```
  FileCheckWait=600
  FileCheckWaitForZero=30
  ```

- When the installation is complete, verify the **Convert** option **prepare.exe** program exists under the **VVCP** installation directory.

- You must enable the **RenderMgtTranslator** service and one or both of the following services:

  - **PreviewService**

  Configure translation services by enabling and configuring translators using Deployment Center.

  - **PreviewService**

    Requires Teamcenter Visualization Convert. Source authoring applications such as Microsoft Office applications are also required.

  - **RenderMgtTranslator**

    Required for either **PreviewService**, **PdfGenerator**, or any other service to be added.

- Use Business Modeler IDE to set up and deploy IRDC and dispatcher service configuration objects to the Teamcenter database.

# 23. Uninstalling Teamcenter

## Uninstall the FOSS Repository

The FOSS Repository provides a central location for all third-party components used by Teamcenter. It is installed automatically in the *TC_ROOT***fossrepo** directory during Teamcenter deployment. However, when uninstalling Teamcenter (deleting the *TC_ROOT* folder), you must take care when deleting the *TC_ROOT***fossrepo** folder.

- If there is only *one* Teamcenter environment on the given machine:

  Delete the FOSS Repository and the **FOSS_REPOSITORY_HOME** environment variable.

  1. Delete the **%TC_ROOT%\fossrepo** and **%TC_ROOT%** directories (on Windows systems) or the **$TC_ROOT/fossrepo** and **$TC_ROOT** directories (on Linux systems).

  2. Remove the **FOSS_REPOSITORY_HOME** system environment variable from the system.

     **Windows Systems:** Delete the environment variable using standard Windows system tools.

     **Linux Systems:**

     a. Remove the definition of **FOSS_REPOSITORY_HOME** from the **$HOME/.profile** file for the user who installed Teamcenter.

     b. Remove the definition of **FOSS_REPOSITORY_HOME** from the **$HOME/.bash_profile** file, if this file exists.

     c. Remove the definition of **FOSS_REPOSITORY_HOME** from the **$HOME/.kshrc** file, if this file exists.

- If there are *multiple* Teamcenter environments on the given machine:

  Do *not* delete the FOSS Repository directory or the **FOSS_REPOSITORY_HOME** environment variable, because the single repository is used by the other installed environments.

### Restoring a deleted FOSS Repository

If the FOSS Repository is mistakenly deleted, you can restore it by performing the following steps:

1. In the Teamcenter software kit, locate the **tc\foss_repository\TpcRepository.zip** file and expand its contents to the path specified by the **FOSS_REPOSITORY_HOME** variable.

2. Open the *FOSS-repository-home***\bin\tpc.bat** file (on Windows systems) or *FOSS-repository-home***/bin/tpc.sh** file (on Linux systems) in a text editor.

3. Replace **@CHANGE_ME_JAVA_HOME@** with the location of the Java installation. Save the changes to the file.

4. In an administrator command prompt, type the following command:

   **Windows Systems:**

   **%FOSS_REPOSITORY_HOME%\bin\tpc.bat ImportProduct** *Tc-kit-location***\tc\foss_repository\TcPackage_wntx64.zip**

   **Linux Systems:**

   **$FOSS_REPOSITORY_HOME/bin/tpc.bat ImportProduct** *Tc-kit-location***/tc/foss_repository/TcPackage_wntx64.zip**

   Replace *Tc-kit-location* with the path to the Teamcenter software kit.

## Uninstall TCCS

If you installed Teamcenter client communication system (TCCS) as part of an installation of the rich client or Teamcenter Microsoft Office interfaces, uninstalling those clients automatically uninstalls TCCS from your system.

If you installed TCCS using the stand-alone installation wizard, perform the following steps to uninstall TCCS.

1. Stop the FMS client cache (FCC) process:

   a. Open a command prompt.

   b. Change to the **\tccs\bin** directory in the TCCS installation directory.

      The default TCCS installation directory is **C:\Program Files\Siemens\Teamcenter***version***\tccs**.

   c. Type the following command:

      ```
      fccstat -stop
      ```

      After stopping the FCC process, the **fccstat** command reports that the FCC is offline.

   d. Close the command prompt.

2. Uninstall TCCS:

   a. In the Windows Control Panel, open the **Add or Remove Programs** dialog box.

b. In the list of installed programs, select and remove **Teamcenter client communication system**.

c. Restart the system to unset the **FMS_HOME** environment variable.

1. Stop the FMS client cache (FCC) process:

   a. Change to the **bin** directory in the TCCS installation directory.

   b. Type the following command to stop the FCC process:

      ```
      fccstat -stop
      ```

2. Change to the **_uninst** directory in the TCCS installation directory.

3. Type the following command:

   ```
   uninstaller.bin
   ```

   This launches the TCCS uninstallation wizard. Follow the instructions in the wizard to uninstall TCCS.

4. Log off and log back on to the system to unset the **FMS_HOME** environment variable.

## Uninstall database software

Uninstall your database software (Oracle or Microsoft SQL Server) according to the vendor documentation.

# 24. Application names changed in Deployment Center

As Active Workspace applications were combined into the Teamcenter application tree in Deployment Center, some application names under the Active Workspace software were changed to help with identification and clarify navigation. The following table shows the old names of these applications with their new names as displayed in Deployment Center 2412.

Note that package IDs and template names have not changed, so the changed display names have no impact on Quick Deploy scripts.

| Old Name | New Name | Package ID |
|---|---|---|
| Aerospace and Defense Change Management | Aerospace and Defense Change Management for Active Workspace | adc1awadschangemanagement |
| Aerospace and Defense Foundation | Aerospace and Defense Foundation for Active Workspace | ads1awadsfoundation |
| Brand Management | Brand Management for Active Workspace | brm1brndmgmtaw |
| Capital Asset Lifecycle Management | Capital Asset Lifecycle Management for Active Workspace | pdm1plantdatamgmtaw |
| Change Management | Change Management for Active Workspace | Cm1cmaws |
| Contract Data Management | Contract Data Management for Active Workspace | cdm1awcontractmanagement |
| EDA Server Support | EDA Server Support for Active Workspace | eda1edaserveraw |
| Embedded Software Management | Embedded Software Management for Active Workspace | esw1esmgmtaw |
| Engineering Change Processes | Engineering Change Processes for Active Workspace | ec1engchangeaw |
| Feature Planning | Feature Planning for Active Workspace | pca1awconfigurator |
| Finish Management | Finish Management for Active Workspace | fsh1awfinishmanagement |
| Initiative Lifecycle Management Overlay for Semiconductor Solution | Initiative Lifecycle Management Overlay for Semiconductor Solution for Active Workspace | ips1scipmoverlayaw |
| Library Management | Library Management for Active Workspace | lbr1librarymgmtaw |

| Old Name | New Name | Package ID |
|---|---|---|
| Material Management | Material Management for Active Workspace | mtw0materialmgmtaw |
| Packaging and Artwork | Packaging and Artwork for Active Workspace | pka1pkgartaw |
| Part Manufacturing | Part Manufacturing for Active Workspace | pm1partmanufacturingaw |
| Partitions for Structure | Partitions for Structure for Active Workspace | ptn0awpartitionforstructure |
| Product Configurator | Product Configurator for Active Workspace | pca0awconfigurator |
| Product Planning | Product Planning for Active Workspace | pgp1awprgplanningapp |
| RAMS Modeling | RAMS Modeling for Active Workspace | ramsmodeling_awclient |
| Relationship Browser | Digital Thread Navigation | relationshipviewer |
| Requirements Management foundation | Requirements Management foundation for Active Workspace | arm0activeworkspacereqmgmt |
| Simulation Process Management | Simulation Process Management for Active Workspace | cae1caeaws |
| Stock Material | Stock Material for Active Workspace | sm1awstockmaterial |
| System Modeling Integration | System Modeling Integration for Active Workspace | umlsysml_awclient |
| Vendor Management | Vendor Management for Active Workspace | vm1vendormanagementaw |
| Work Package Management | Work Package Management for Active Workspace | wpm1awpkgmgmt |

# 25. Security Services properties in Deployment Center

The following tables map Security Services context parameters to Deployment Center properties for Security Services web applications.

If you previously installed Security Services and built the Login Service and Identity Service web applications using the Web Application Manager (**insweb**), you set these context parameters in that tool.

You can alternatively build the **Login Service** and **Identity Service** web applications using Deployment Center. Security Services context parameters in the Web Application Manager map to Security Services properties In Deployment Center.

### Properties for the Security Services Login Service

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **webmaster** | **fnd0_tcSSEmailAddress** | Web Master | The email address of the administrator to whom questions and comments about this application should be addressed. |
| **tcsso.login_service.appid** | **fnd0_tcSSOApplicationId** | Login Service Application ID | The Teamcenter ID of the Security Services Login Service Web Application. This should match the corresponding entry in the Application Registry Table of the Identity Service. |
| **tcsso.login_service.rp_cookieNamePattern** | **fnd0_tcSSOCookieNamePattern** | Cookie Name Pattern | A pattern or set of patterns describing the names of cookies used by Reverse Proxy Servers protecting Teamcenter applications |
| **tcsso.login_service.proxyURL** | **fnd0_tcSSOLoginProxyURL** | Login Service Proxy URL | The **protocol://host:port** URL for the Teamcenter Security Services Login Service when used with load balancing or Commercial SSO proxies. |
| **tcsso.login_service.sso_service_url** | **fnd0_tcSSServiceURL** | Identity Service URL | Identity Service URL |
| **identityServicePassword** | **fnd0_identity_service_password** | Identity Service Password | Specifies a password that the Identity Service hashes to form keys for signing and encrypting security information to prevent it from being forged or viewed. Security improves with the length and randomness of the |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| | | | password. This is used for auto-login and commercial SSO and MUST match the same parameter in the TcSS Login Service. The value for this parameter will be encrypted. |
| **tcsso.behind_sso_gateway** | **fnd0_tcSSOGateway** | Is the Login Service behind a Gateway (Gateway mode)? | This flag indicates the presence of a third-party single sign-on solution |
| **tcsso.gateway.field.type** | **fnd0_tcSSOGatewayfieldType** | Gateway Field Type | This string indicates how the gateway will transmit credential information (Teamcenter User ID) in the HTTP request to the Login Service. |
| **tcsso.gateway.field.name** | **fnd0_tcSSOGatewayfieldName** | Gateway Field Name | A string value that is the name of the **tcsso.gateway.field.type** field in the settings file. Allowed values are: "Header", "Cookie","Principal", "Remote_User", "Client_Certificate" and "Filter_Class". This value is ignored if **tcsso.behind_sso_gateway** is false. |
| **tcsso.gateway.logout_url** | **fnd0_tcSSOGatewayLogoutURL** | Gateway logout URL | If set, the TcSS LoginService will redirect to this URL to logout of the gateway session. |
| **tcsso.username.filter.class** | **fnd0_tcSSOGatewayfilterUserName** | Gateway Filter User Name | This value will only be used when **tcsso.gateway.field.type** value is filter_class. |
| **tcsso.client.enable.notice.consent.logon.banner** | **fnd0_tcSSOLogonBanner** | Notice and Consent Log-on Banner | A boolean value that represents if notice and consent log-on banner should be displayed to user. |
| **tcsso.forgotten.password.URL** | **fnd0_tcSSOForgottenPasswordURL** | URL to reset the forgotten password | If it is non-empty and is a valid URL, it is the value associated with the Forgot password hypertext link, which will then appear on the login page. If empty, the Forgot password link will not appear. It is assumed the URL references a page where the user can request a new password or that their old password be sent. |
| **tcsso.online_help.enable** | **fnd0_tcSSOOnlineHelp** | Online Help Enable | This flag enables/disables Security Services online help. |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| | | | If true, the online help is available to users. If false, it becomes unavailable. |
| **tcsso.login_service.force_web_browser_login** | **fnd0_tcSSOWebBrowserLogin** | Login Service Force Web Browser Login | This flag disables single sign-on among browser instances on the user's workstation. |
| **tcsso.frame_ancestors** | **fnd0_tcSSOFrameAncestors** | Frame Ancestors | Indicates whether a browser should be allowed to render the login page in a frame or iframe. Use this to avoid "clickjacking" attacks by ensuring the login page is not embedded. |
| **Log Level** | **fnd0_tcSSOLogLevel** | Log Level | Specify the log level for the Login Service Logger |
| **Log File** | **fnd0_tcSSOLogFile** | Logger File Name | Specify the name of the log file used for the Login Service Logger. |
| **tcsso.federation_type** | **fnd0_tcSSOFederationType** | Federation Type | If set, the TCSS LoginService will rely on the Federation Identity Provider to perform user authentication and TCSS will perform as service provider and authorize users for Teamcenter applications. |
| **tcss.federation_url** | **fnd0_tcSSOFederationURL** | Federation URL | If set,the TCSS LoginService redirects the user to the URL for the Federation Identity Provider to perform user authentication and return the user to the Teamcenter application after successful authentication (*blank*).<br>The URL needs to be a single sign-on (SSO) URL that conforms to the Security Assertion Markup Language (SAML) standard, that is, a service provider-initiated login URL provided by the SAML identity provider (IdP). |
| **tcsso.federation_reply_url** | **fnd0_tcSSOFederationReplyURL** | Federation Reply URL | If set, the TcSS Login Service will provide this URL to the Federation Identity Provider to redirect user to after authentication. |
| **tcsso.federation_logout_url** | **fnd0_tcSSOFederationLogoutURL** | Federation Logout URL | If set, the TCSS Login Service will provide this URL to the Federation Identity Provider to logout the user from both the Identity and Service Providers. |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **tcsso.samauth.client_id** | **fnd0_tcsso.samauth.client_id** | SAM Authorization client ID | SAM Authorization client ID |
| **tcsso.samauth.client_secret** | **fnd0_tcsso.samauth.client_secret** | SAM Authorization client Secret | SAM Authorization client Secret |
| **tcsso.samauth.auth_endpoint** | **fnd0_tcsso.samauth.auth_endpoint** | SAM Authorization Endpoint | SAM Authorization Endpoint |
| **tcsso.samauth.token_endpoint** | **fnd0_tcsso.samauth.token_endpoint** | SAM Authorization token Endpoint | SAM Authorization token Endpoint |
| **tcsso.samauth.jwks_endpoint** | **fnd0_tcsso.samauth.jwks_endpoint** | SAM Authorization jwks Endpoint | SAM Authorization jwks Endpoint |
| **tcsso.samauth.userid_claim** | **fnd0_tcSSSamAuthUserId** | SAM Authorization Claim User ID | SAM Authorization Claim User ID |
| **tcsso.saml.issuer_id** | **fnd0_tcsso.saml.issuer_id** | SAML Issuer ID | SAML Issuer ID |
| **tcsso.saml.idp_public_key_file** | **fnd0_tcsso.saml.idp_public_key_file** | SAML IDP Public Key File | SAML IDP Public Key File |
| **tcsso.saml.decryption_ private_jks_file** | **fnd0_tcsso.saml.decryption_private_ jks_file** | Decryption private JKS File Name | Decryption private JKS File Name |
| **tcsso.saml.decryption_ private_jks_file_pwd** | **fnd0_tcsso.saml.decryption_private_ jks_file_pwd** | Decryption private JKS File Password | Decryption private JKS File Password |
| **tcsso.saml.decryption_ private_key_name** | **fnd0_tcsso.saml.decryption_private_ key_name** | Decryption private key Name | Decryption private key Name |
| **tcsso.saml.decryption_ private_key_pwd** | **fnd0_tcsso.saml.decryption_private_ key_pwd** | Decryption private key Password | Decryption private key Password |
| **tcsso.saml.signing_ private_jks_file** | **fnd0_tcsso.saml.signing_private_ jks_file** | Signing private JKS File Name | Signing private JKS File Name |
| **tcsso.saml.signing_ private_jks_file_pwd** | **fnd0_tcsso.saml.signing_private_ jks_file_pwd** | Signing private JKS File Password | Signing private JKS File Password |
| **tcsso.saml.signing_ private_key_name** | **fnd0_tcsso.saml.signing_private_ key_name** | Signing private key Name | Signing private key Name |
| **tcsso.saml.signing_ private_key_pwd** | **fnd0_tcsso.saml.signing_private_ key_pwd** | Signing private key Password | Signing private key Password |
| **tcsso.oidc.client_id** | **fnd0_tcsso.oidc.client_id** | OIDC Authorization client ID | OIDC Authorization client ID |
| **tcsso.oidc.client_secret** | **fnd0_tcsso.oidc.client_secret** | OIDC Authorization client Secret | OIDC Authorization client Secret |
| **tcsso.oidc.auth_endpoint** | **fnd0_tcsso.oidc.auth_endpoint** | OIDC Authorization Endpoint | OIDC Authorization Endpoint |
| **tcsso.oidc.token_endpoint** | **fnd0_tcsso.oidc.token_endpoint** | OIDC Authorization token Endpoint | OIDC Authorization token Endpoint |
| **tcsso.oidc.jwks_endpoint** | **fnd0_tcsso.oidc.jwks_endpoint** | OIDC Authorization jwks Endpoint | OIDC Authorization jwks Endpoint |
| **tcsso.oidc.userid_claim** | **fnd0_tcSSoidcUserId** | OIDC Authorization User ID | OIDC Authorization User ID |
| **tcsso.oidc.signing_jks_file** | **fnd0_tcsso.oidc.signing_jks_file** | OIDC Signing File Name | OIDC Signing File Name |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **tcsso.oidc.signing_jks_file_pwd** | **fnd0_tcsso.oidc.signing_jks_file_pwd** | OIDC Signing file Password | OIDC Signing file Password |
| **tcsso.oidc.signing_private_key_name** | **fnd0_tcsso.oidc.signing_private_key_name** | OIDC Signing private key Name | OIDC Signing private key Name |
| **tcsso.oidc.signing_private_key_pwd** | **fnd0_tcsso.oidc.signing_private_key_pwd** | OIDC Signing private key Password | OIDC Signing private key Password |
| **tcsso.oidc.encryption_jks_file** | **fnd0_tcsso.oidc.encryption_jks_file** | OIDC Encryption jks File Name | OIDC Encryption jks File Name |
| **tcsso.oidc.encryption_jks_file_pwd** | **fnd0_tcsso.oidc.encryption_jks_file_pwd** | OIDC Encryption jks file Password | OIDC Encryption jks file Password |
| **tcsso.oidc.encryption_private_key_name** | **fnd0_tcsso.oidc.encryption_private_key_name** | OIDC Encryption key Name | OIDC Encryption key Name |
| **tcsso.oidc.encryption_private_key_pwd** | **fnd0_tcsso.oidc.encryption_private_key_pwd** | OIDC Encryption key Password | OIDC Encryption key Password |
| **tcsso.cors_whitelist** | **fnd0_tcSSOCORSSupport** | CORS Support Domains | If set, the list of domains to be white-listed are honored for CORS support. |
| **tcsso.login_service.enableCsrf** | **fnd0_tcSSOLoginServiceEnableCSRF** | Enable Session Agent CSRF Protection | Set "true" to enable CSRF protection in Session Agent communications with the Login Service. Default is false to allow compatibility with clients on older releases. |
| **tcsso.login_service.csrf.cookie.httpOnly** | **fnd0_tcSSOCSRFCookieHttpOnly** | CSRF Cookie http Only | Set "true" to enable httponly flag on CSRF cookie. Default is true. |
| **tcsso.login_service.csrf.cookie.SameSite_None** | **fnd0_tcSSOCSRFCookieSameSiteNone** | CSRF Cookie Same Site None Flag | Set "true" to enable SameSite=None flag on CSRF cookie. Default is false. |
| **tcsso.login_service.umcsession.cookie.SameSite_None** | **fnd0_tcSSOCSRFCookieUMCSession** | UMC Session Cookie Same Site None Flag | Set "true" to enable SameSite=None flag on umcsession cookie. Default is false. |
| **tcsso.login_service.session_cookie_name** | **fnd0_tcSSOSessionCookieName** | Servlet Session Cookie Name | Specify a custom Servlet session cookie name. If this parameter is not set, TcSS-JSESSIONID will be used by default. |
| **tcsso.login_service.session_cookie_path** | **fnd0_tcSSOSessionCookiePath** | Servlet Session Cookie Path | Specify a custom Servlet session cookie path value. If this parameter is not set, the Servlet context path will be used by default. For example, if the Login Service web app name is LoginService, the default session cookie path will be LoginService. |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **tcsso.login_service.session_cookie_httponly** | **fnd0_tcSSOSessionCookiehttponly** | Session Cookie http Only | Specify whether or not the Servlet session cookie includes the HttpOnly flag. |
| **tcsso.login_service.session_cookie_secure** | **fnd0_tcSSOSessionCookieSecure** | Servlet Session Secure Flag | Specify whether or not the Servlet session cookie includes the Secure flag |
| **propertiesPassword** | **fnd0_tcSSPropertiesPassword** | propertiesPassword | **fnd0_tcSSPropertiesPassword** |
| **DEBUG** | *Deprecated in Web Application Manager, no property in Deployment Center* | DEBUG | *Deprecated in Web Application Manager, no property in Deployment Center* |
| **tcsso.loginInputDef_name_*,**<br>**tcsso.loginInputDef_where_*,**<br>**tcsso.loginInputDef_required_*,**<br>**tcsso.loginInputDef_idpkey_*** | **fnd0_loginInputDefinitions** | Login Input Definition Table | Login Input Definition Table |
| **tcsso.loginInputDef_name_1** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_name_2** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_name_3** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_name_4** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_name_5** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_where_1**<br>**tcsso.loginInputDef_where_2**<br>**tcsso.loginInputDef_where_3**<br>**tcsso.loginInputDef_where_4**<br>**tcsso.loginInputDef_where_5** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_required_1**<br>**tcsso.loginInputDef_required_2** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_required_3**<br>**tcsso.loginInputDef_required_4**<br>**tcsso.loginInputDef_required_5** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_idpkey_1** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_idpkey_2** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_idpkey_3** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_idpkey_4** | **fnd0_loginInputDefinitions** | | |
| **tcsso.loginInputDef_idpkey_5** | **fnd0_loginInputDefinitions** | | |
| **session-timeout** | **fnd0_tcSSSessionTimeout** | Session Timeout | |
| **tcsso.samauth.scope** | **fnd0_tcsso.samauth.scope** | SAM Authorization Scope | SAM Authorization Scope |
| **tcsso.saml.validate.response.signature** | **fnd0_tcsso.saml.validate.response.signature** | SAML Signature | SAML Signature |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **tcsso.saml.want.assertion.encrypted** | **fnd0_tcsso.saml.want.assertion.encrypted** | SAML Assertion Encryption | SAML Assertion Encryption |
| **tcsso.saml.sign_authn_request** | **fnd0_tcsso.saml.sign_authn_request** | SAML Authentication Request | SAML Authentication Request |
| **tcsso.saml.userid_attribute_name** | **fnd0_tcsso.saml.userid_attribute_name** | SAML User ID | SAML User ID |
| **tcsso.oidc.scope** | **fnd0_tcsso.oidc.scope** | OIDC Authorization Scope | OIDC Authorization Scope |
| **tcsso.oidc.client_auth_method** | **fnd0_tcsso.oidc.client_auth_method** | OIDC Client Authentication Method | OIDC Client Authentication Method |
| **tcsso.oidc.jwt.sig_alg** | **fnd0_tcsso.oidc.jwt.sig_alg** | Private key jwt Signing Algorithm | Private key jwt Signing Algorithm |
| **tcsso.oidc.jwt.expiration** | **fnd0_tcsso.oidc.jwt.expiration** | Private key jwt expiration Time | Private key jwt expiration Time |

## Properties for the Security Services Identity Service

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **webmaster** | **fnd0_tcSSEmailAddress** | Web Master | The email address of the administrator to whom questions and comments about this application should be addressed. |
| **LDAPVersion** | **fnd0_tcss_ldap_version** | LDAP Version | Sets the minimum LDAP version used for connections. |
| **PasswordResetEnabled** | **fnd0_identity_password_reset_enabled** | Enable Password Reset? | If true, detection of a reset or expiration of password will result in the Login Page displaying the password reset fields. If false, the user will receive the Password is incorrect message. This capability is currently available for Active Directory only. |
| **PasswordResetMessage** | **fnd0_identity_password_reset_message** | Password Reset Message | Additional information displayed to user when prompted to change password. This may be a link to a change password service. |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **GatewayAliasingEnabled** | **fnd0_identity_gateway_ aliasing_enabled** | Enable Gateway Aliasing? | User ID aliasing for Teamcenter applications is always performed unless TcSS is configured in gateway or TcSSAutoLogin mode. If this parameter is 'true', User ID aliasing is performed in gateway and TcSSAutoLogin modes as well. Unless a valid LDAP repository is configured in TcSS, this parameter must be set to 'false'. |
| **ReferralsEnabled** | **fnd0_identity_referrals_enabled** | Enable Referrals? | If true, LDAP referrals are followed across LDAP servers. Otherwise, referrals are ignored. |
| **ReferralHopLimit** | **fnd0_identity_referral_hoplimit** | Referral Hop Limit | Sets the maximum number of hops to follow in sequence during a referral. This value is ignored if ReferralsEnabled is false. |
| **LDAPIdleConnectionTimeout** | **fnd0_tcss_ldap_connection_ timeout** | LDAP Connection Timeout | Sets the idle timeout (in minutes) for cached LDAP connections in the pool. Connections that are idle in the connection pool for this period will be purged from the pool. Setting this value to 0 specifies infinite timeout (connections are only purged if there is a connection failure). |
| **DEBUG** | *Deprecated in Web Application Manager, no property in Deployment Center* | DEBUG | *Deprecated in Web Application Manager, no property in Deployment Center* |
| **identityProvider** | **fnd0_identity_provider** | Identity Provider | Specifies the interface class to the Identity Provider used by the Single Sign-On Service. The default class provides integration with any LDAP v3-compliant directory service. A customer-supplied class implementing the Identity Provider |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| | | | interface can also be specified here. |
| **identityServicePassword** | **fnd0_identity_service_password** | Identity Service Password | Specifies a password that the Identity Service hashes to form keys for signing and encrypting security information to prevent it from being forged or viewed. Security improves with the length and randomness of the password. This is used for auto-login and commercial SSO and MUST match the same parameter in the TSS Login Service. The value for this parameter will be encrypted. |
| **passwordLifetime** | **fnd0_identity_password_lifetime** | Password Life Time | Lifetime, in seconds, for auto-login or commercial SSO attempt. This time will limit a replay attack. This is configurable to accommodate latency in deployments. |
| **mediatorPassword** | **fnd0_identity_mediator_password** | Mediator Password | A password shared between the Identity Service and a Mediating Application. Used to encrypt tokens passed to the Mediator for later distribution to applications participating in Trust Relationships. The value for this parameter will be encrypted. |
| **tokenLifetime** | **fnd0_identity_token_lifetime** | Token Life Time | Lifetime, in seconds, of a Single Sign-On token. |
| **sessionLifetime** | **fnd0_identity_session_lifetime** | Session Life Time | Maximum time, in minutes, that a Single Sign-On session can be idle before expiring. It must be greater than or equal to 1 and less than or equal to 60000. |
| **Log Level** | **fnd0_identity_log_level** | Log Level | Specify the log level for the Identity Service Logger. |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| **Log File** | **fnd0_identity_log_file** | Log File | Specify the name of the log file used for the Identity Service Logger. |
| **tcsso.LogLevel** | *Deprecated in Web Application Manager, no property in Deployment Center* | | |
| **tcsso.AuthLogDir** | *Deprecated in Web Application Manager, no property in Deployment Center* | | |
| **OAuthTokensEnabled** | **fnd0_identity_OAuth_tokens_ enabled** | Enable OAuth Tokens? | Specifies whether or not OAuth Access Tokens are accepted as Teamcenter login credentials. |
| **propertiesPassword** | **fnd0_tcSSPropertiesPassword** | Properties Password | Specifies the password that will be used to decrypt encrypted property values<br>stored in properties files (for example, **federation.properties**). The value for this parameter will be encrypted. |
| **tcsso.DomainMap_UserDomain_1** | **fnd0_tcss_ldapDomainMap** | LDAP Domain MAP Table | LDAP Domain MAP Table |
| **tcsso.DomainMap_LDAPDomain_1** | **fnd0_tcss_ldapDomainMap** | | |
| **LDAPConfiguration_LDAPOrdinal_1** | **fnd0_tcss_ldapOrdinal** | LDAP Configuration Table | |
| **LDAPConfiguration_LDAPHost_1** | **fnd0_TcSS_LDAP.dc0_machine_ name** | | |
| **LDAPConfiguration_LDAPPortNo_1** | **fnd0_tcss_ldapPort** | | |
| **LDAPConfiguration_LDAPPortNo Override_1** | **fnd0_tcss_ldapPortOverride** | | |
| **LDAPConfiguration_LDAPConnectType_1** | **fnd0_tcss_ldapProtocol** | | |
| **LDAPConfiguration_MaxLDAP Connections_1** | **fnd0_tcss_ldap_max_connections** | | |
| **LDAPConfiguration_QueryDN_1** | **fnd0_tcss_ldapAdministratorDN** | | |
| **LDAPConfiguration_QueryDNPassword_1** | **fnd0_tcss_ldapAdministrator Password** | | |
| **LDAPConfiguration_LDAPBaseDN_1** | **fnd0_tcss_ldap_base_dn** | | |
| **LDAPConfiguration_UserObjectClass_1** | **fnd0_tcss_ldapUserObjectClass** | | |
| **LDAPConfiguration_MapUserAttribute_1** | **fnd0_fallBack_userAttr** | | |
| **LDAPConfiguration_UserAttribute_1** | **fnd0_tcss_ldapUserAttribute** | | |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| LDAPConfiguration_LDAPConnection SetupDelay_1 | fnd0_tcss_ldap_connection_ setupDelay | | |
| LDAPConfiguration_LDAPConnect Timeout_1 | fnd0_tcss_ldap_connection_ timeout | | |
| tcsso.applicationRegistry_ApplicationID_1 | fnd0_tcss_appRegistry | Application Registry Table | |
| tcsso.applicationRegistry_ApplicationID_2 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_3 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_4 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_5 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_6 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_7 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_8 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_9 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_10 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_11 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_12 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_ApplicationID_13 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_1 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_2 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_3 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_4 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_5 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_6 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_7 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_8 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_9 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_10 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_11 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_12 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_RootURL_13 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_1 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_2 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_3 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName | fnd0_tcss_appRegistry | | |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| Attr_4 | | | |
| tcsso.applicationRegistry_AppUserName Attr_5 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_6 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_7 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_8 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_9 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_10 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_11 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_12 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_AppUserName Attr_13 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_Trusted_1<br>tcsso.applicationRegistry_Trusted_2<br>tcsso.applicationRegistry_Trusted_3<br>tcsso.applicationRegistry_Trusted_4<br>tcsso.applicationRegistry_Trusted_5<br>tcsso.applicationRegistry_Trusted_6<br>tcsso.applicationRegistry_Trusted_7<br>tcsso.applicationRegistry_Trusted_8<br>tcsso.applicationRegistry_Trusted_9<br>tcsso.applicationRegistry_Trusted_10<br>tcsso.applicationRegistry_Trusted_11<br>tcsso.applicationRegistry_Trusted_12<br>tcsso.applicationRegistry_Trusted_13 | fnd0_tcss_appRegistry | | |
| tcsso.applicationRegistry_StripDomain_1<br>tcsso.applicationRegistry_StripDomain_2<br>tcsso.applicationRegistry_StripDomain_3<br>tcsso.applicationRegistry_StripDomain_4<br>tcsso.applicationRegistry_StripDomain_5 | fnd0_tcss_appRegistry | | |

| Web Application Manager Property | Deployment Center Property Internal Name | Deployment Center Property Display Name | Deployment Center Property Description |
|---|---|---|---|
| tcsso.applicationRegistry_StripDomain_6 | | | |
| tcsso.applicationRegistry_StripDomain_7 | | | |
| tcsso.applicationRegistry_StripDomain_8 | | | |
| tcsso.applicationRegistry_StripDomain_9 | | | |
| tcsso.applicationRegistry_StripDomain_10 | | | |
| tcsso.applicationRegistry_StripDomain_1 | | | |
| tcsso.applicationRegistry_StripDomain_11 | | | |
| tcsso.applicationRegistry_StripDomain_12 | | | |
| tcsso.applicationRegistry_StripDomain_13 | | | |

# 26. Required RPM package managers

If you use the visualization server manager (VSM) on a Linux machine, make sure the following required RPM package managers are available on the machine.

**SUSE Linux:**

**fontconfig-2.11.1-7.1.x86_64**
**glibc-2.31-150300.46.1.x86_64**
**glibc-32bit-2.22-15.3.x86_64**
**libbz2-1-1.0.6-29.2.x86_64**
**libexpat1-2.1.0-21.3.1.x86_64**
**libexpat-devel-2.1.0-21.3.1.x86_64**
**libfreetype6-2.6.3-7.15.1.x86_64**
**libgcc_s1-8.2.1+r264010-1.3.3.x86_64**
**libGLU1-9.0.0-18.1.x86_64**
**libICE6-1.0.8-12.1.x86_64**
**libjpeg8-8.1.2-31.7.4.x86_64**
**libpng16-16-1.6.8-14.1.x86_64**
**libSM6-1.2.2-3.59.x86_64**
**libstdc++6-8.2.1+r264010-1.3.3.x86_64**
**libstdc++6-12.2.1+git416-150000.1.7.1.x86_64**
**libuuid1-2.29.2-7.14.x86_64**
**libX11-6-1.6.2-12.5.1.x86_64**
**libXau6-1.0.8-4.58.x86_64**
**libxcb1-1.10-4.3.1.x86_64**
**libXext6-1.3.2-4.3.1.x86_64**
**libXft2-2.3.1-9.32.x86_64**
**libXm4-2.3.4-4.15.x86_64**
**libXmu6-1.1.2-3.60.x86_64**
**libXp6-1.0.2-3.58.x86_64**
**libXrender1-0.9.8-7.1.x86_64**
**libXt6-1.1.4-3.59.x86_64**
**libz1-1.2.11-1.27.x86_64**
**Mesa-libGL1-18.0.2-6.28.x86_64**

> Note:
>
> On SUSE Linux, the **/usr/lib64/libGLdispatch.so.0** file is not owned by any package.
>
> Also, the **/usr/lib64/libGLX.so.0** file is not owned by any package.

**RedHat Linux:**

**bzip2-libs-1.0.6-13.el7.x86_64**
**expat-2.1.0-10.el7_3.x86_64**
**expat-devel-2.1.0-10.el7_3.x86_64**

**fontconfig-2.13.0-4.3.el7.x86_64**
**freetype-2.8-12.el7.x86_64**
**glibc-2.28-225.el8.x86_64**
**libgcc-4.8.5-36.el7.x86_64**
**libglvnd-1.0.1-0.8.git5baa1e5.el7.x86_64**
**libglvnd-glx-1.0.1-0.8.git5baa1e5.el7.x86_64**
**libICE-1.0.9-9.el7.x86_64**
**libjpeg-turbo-1.2.90-6.el7.x86_64**
**libpng-1.5.13-7.el7_2.x86_64**
**libSM-1.2.2-2.el7.x86_64**
**libstdc++-4.8.5-36.el7.x86_64**
**libstdc++-8.5.0-18.el8.x86_64**
**libuuid-2.23.2-59.el7.x86_64**
**libX11-1.6.5-2.el7.x86_64**
**libXau-1.0.8-2.1.el7.x86_64**
**libxcb-1.13-1.el7.x86_64**
**libXext-1.3.3-3.el7.x86_64**
**libXft-2.3.2-2.el7.x86_64**
**libXmu-1.1.2-2.el7.x86_64**
**libXp-1.0.2-2.1.el7.x86_64**
**libXrender-0.9.10-1.el7.x86_64**
**libXt-1.1.5-3.el7.x86_64**
**mesa-libGLU-9.0.0-4.el7.x86_64**
**motif-2.3.4-14.el7_5.x86_64**
**zlib-1.2.7-18.el7.x86_64**