TEAMCENTER

Deployment Center — Usage

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

Quick deployment

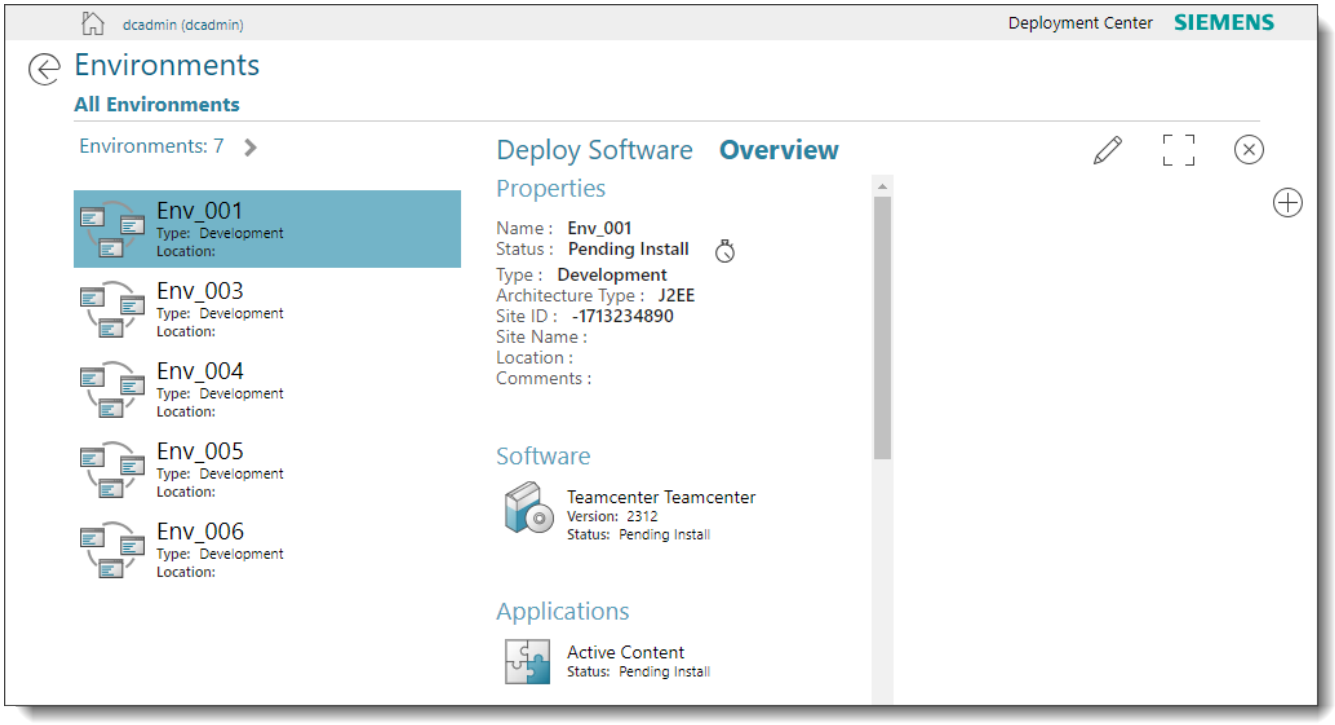
How to deploy the Business Modeler IDE templates on Teamcenter

1. Introduction to Deployment Center

Getting started with Deployment Center

Deployment Center is a centralized web application that simplifies the deployment, management, and maintenance of Teamcenter software. It supports creation of multiple Teamcenter environments and generation of deployment scripts to install or update software on each machine in an environment.

Deployment Center is a centralized web application for deploying software to Teamcenter environments. Deployment Center gives you access to multiple environments from a single, centralized location. It simplifies the process of installing and updating software and can be used with third-party automation tools. You can manage a variety of Teamcenter environments more efficiently.



Download software packages to a central location, set up your installation or update in Deployment Center, and generate deployment scripts. Run the scripts on target machines that access and install the software.

Benefits

- Deploy, manage, and maintain multiple versions of software in multiple Teamcenter environments from a single web application with authenticated user access.
- Reduce the costs of managing Teamcenter environments by reducing the number of people, steps, and time involved in software deployment.

- Provide a centralized location to create and view Teamcenter environments and view deployment information.

- Ensure consistency and accuracy by reviewing and adjusting deployment configuration before rollout to Teamcenter environments.

- When setting up your environment, take advantage of entering parameters for one component and having those values shared with related components. For example, when you enter a machine name or port number for a component, that information is shared with other components that require the same information.

Flexible administration of software deployment

The software deployment process is accomplished in two phases.

1. Select the software and configure the deployment in the Deployment Center web application. You can manage responsibility for software deployment using a single administrator or share it among the site's deployment experts. For example:

   - A business analyst can make decisions about choosing the applications. Business analysts can choose the software versions and the applications needed by business units.

   - An administrator can make decisions about the machines, ports, URLs, user names, passwords, and component settings for the target machines in an environment. Administrators can make deployment selections based on knowledge of hardware infrastructure, number of users, and which applications are used by business units. For more information about recommended distribution of hardware and components, see the *Teamcenter Deployment Reference Architecture*, which provides detailed examples of Teamcenter deployments, plus sample deploy scripts and other resources. This package is available in the **Teamcenter Downloads area on Support Center**.

2. After reviewing the deployment configuration, generate the deployment scripts and deploy the software on the target machines.

Manage software deployment for Teamcenter environments

You can deploy a variety of applications and components among your Teamcenter environments. A **typical deployment process** specifies software configuration and generates deployment scripts that install or update the software on target servers using software kits from a common repository.

Manage software deployment using Deployment Center utilities

You can deploy Teamcenter environments by following a **quick deployment process** that uses a Deployment Center utility to manage application and component configurations and generate deployment scripts.

Deployment Center key concepts

The following terms and concepts are used throughout Deployment Center.

- **Environment**

  Centrally-managed environment you model in Deployment Center to deploy and update Teamcenter.

- **Software**

  - Deployment Center-compatible software kits downloaded from Support Center.
  - Custom Deployment Center-compatible software packages (automatically created by Business Modeler IDE, etc.).

- **Applications**

  Related binaries, files, and data model that comprise a set of Teamcenter functionality, contained in a Deployment Center-compatible software package.

- **Components**

  Logical representation of Teamcenter infrastructure components.

Finding information

The following Deployment Center documentation deliverables describe how to create, deploy, and manage Teamcenter environments using the Deployment Center application and its utilities:

- *Deployment Center — Usage* (This deliverable)
- *Teamcenter Installation Using Deployment Center*
- *Teamcenter Deployment Reference Architecture*

To learn how to deploy a specific solution, application, or server component, see the documentation for the given version that you want to install or update.

Deployment considerations

System requirements and compatibility

Download support documents from Support Center to find system software certified for Deployment Center and supported upgrade paths for Deployment Center and Teamcenter.

Limitations

- See the Teamcenter Interoperability Matrix (**Teamcenter Interoperability Matrix** *date*.**xlsx**) for information about versions of Deployment Center that can be upgraded directly to Deployment Center 2412.
- To perform deployment actions that are not yet available in Deployment Center, run Teamcenter Environment Manager (TEM) on the target machines to complete your environment updates and configuration.

For additional information specific to this release, see the Deployment Center README document available with the Deployment Center 2412 product download.

Deployment process

Using Deployment Center, you can create and configure Teamcenter environments, with the flexibility to change applications, components, and parameter settings before you deploy Teamcenter software to the target machines in your network. The deployment process allow you to:

- Import environments created using Teamcenter Environment Manager (TEM) into Deployment Center.
- Register imported environments and further modify them in Deployment Center.
- Add Teamcenter software kits from updates, patches, and asynchronous product releases.
- Review selected applications, components, and parameter settings before you generate deployment scripts.
- Validate deployment scripts before deploying.

In the deployment process, you will perform the following common Deployment Center procedures:

Start Deployment Center

Start and log on to Deployment Center. You can **display information about all existing, registered Teamcenter environments**, including deployed software, applications, server components, and environment properties.

Set up the software repository

Plan where to store Teamcenter and related software kits for deployment.

Software storage considerations

- Keep the software in a single local repository on the Deployment Center web server. Use this option if you don't have spare server space located near target servers.

- Put software in multiple locations that may be shared with your target environment. Plan for enough disk space to hold the entire set of software for an environment. If you have existing software kits in a *software* directory, you can map a UNC path to that location. Be sure the server is accessible to Deployment Center. Use this option to make access easier during deployment.

- Copy the software from the repository to each target server, ensuring each server gets the complete set of software. Specify this path in the **-softwareLocation** argument when you deploy. Plan for additional disk space on each target server. Use this option to install software more quickly.

Manage the repositories

- Specify the software **repository locations** when you install or upgrade Deployment Center. If you already have software on another server, you can include it.

- Deployment Center scans repository locations regularly for unzipped software. The scanned software appears on the **Software Repositories** page which lists all available, valid software kits. If you have software dependencies, the repository notes missing software.

- Place the downloaded software kits in a repository *software* directory. Be sure they have been unzipped.

Manage environments

You can create and manage registered environments in Deployment Center. You can view the list of Teamcenter environments on the **Environments page**.

You can **create a new environment**, **register an existing Teamcenter environment**, or **remove an environment**.

Review registered servers

You can **view all machines** from your registered Teamcenter environments. On the **Machines** page, you can see all server machines used by the components in your registered environments.

Install or update software and applications

In Deployment Center, determine your approach for **installing or updating an environment**. Then **deploy your software** using the Deployment Center application.

You can deploy software in a multiple server environment or on a single machine. If your components are on multiple servers, verify the server configuration for each component.

Configure server components

In Deployment Center, you **configure server components** that are associated with your selected software and applications. Some components may be automatically selected, and some settings may already be configured. You can choose multiple servers for deployment.

Because selections and settings are saved as you go, you can review and verify them. If you are unsure of your setup or you need to make changes, return later to complete or change your settings before deployment.

When you enter user names and passwords for server components in Deployment Center, passwords are encrypted in the generated deployment scripts or you can provide an encrypted password.

Validate deployment scripts

After you verify your configuration information, **generate the deployment scripts**, and copy each script to its corresponding target server. Each script contains the designated target machine name, what to install or update on the target machine, and the software configurations you specified in Deployment Center.

> Tip:
>
> Validate the deployment scripts in a test run. As a best practice, **run the deployment scripts in diagnostic mode** on each target machine to validate the configuration. Check the log output for errors. No updates are made to the target machine during diagnostic validation. Make the corrections in Deployment Center, regenerate the deployment scripts, and run them in diagnostic mode until all errors are addressed.

Deploy the software

After validation is successful, **run the deployment scripts** on each target machine. The deployment script has a parameter to specify the location of the *software* directory. Be sure the software is available to the target machines.

Update or maintain your environments

You can **install or update additional applications and components** in Teamcenter environments managed from Deployment Center. You can also **perform maintenance on component configuration or parameter values** after installation from Deployment Center.

To perform additional deployment actions that are not yet available in Deployment Center, run Teamcenter Environment Manager (TEM) on the target servers to complete your environment updates and configuration.

Password encryption

You can provide an encrypted password for Deployment Center installation, as well as for command line utilities, by creating a secure encrypted password or password file. The **dc_encrypt.bat** utility (on Windows systems) and **dc_encrypt.sh** utility (on Linux systems) are located in the *deployment_center* directory of the software download kit.

When creating an encrypted password, the input password is specified either in text form or in an environment variable that contains the password.

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:

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

Some of these characters are used by operating systems (such as Windows or Linux) and by Java as delimiters. If you use any of these characters for encryption, you must precede them with the appropriate escape characters. For example, escape a backslash character (****) with a backslash (****).

-password

Specify the clear text password for the Deployment Center administrator user that you want to encrypt.

-env

Specify the environment variable containing the Deployment Center administrator password.

-file

Specify the file in which you want to save the encrypted password.

If the path contains spaces, enclose it in quotes.

-help

Displays the usage information for **dc_encrypt**.

Example:

To produce an encrypted password and save it to a file:

```
dc_encrypt -password=dc_password
  -file=D:\pwd_files\dc_admin.pwf
```

To produce an encrypted password without saving to a file:

dc_encrypt -password=*dc_password*

To produce an encrypted password from a password contained in an environment variable:

dc_encrypt -env=*DC_PWD_ENVAR*

Note:

If you generate an encrypted password without using the **-file** argument, you must prefix the encrypted password with **dc:** and save it to a text file before you can use the encrypted password with a Deployment Center utility. For example, save the following to a file named **dc_admin.pwf**:

dc:*encrypted-PW*

You can create an encrypted password or password file that can be used by these Deployment Center utilities:

deployment_center.bat -maintenance can also update an existing password later.

send_configuration_to_dc

deploy

dc_quick_deploy can also **generate encrypted passwords**.

Use **dc_encrypt** to generate password files.

HTTPS communication

Deployment Center supports HTTPS and HTTP communication. HTTPS ensures encrypted, secure communication between the Deployment Center server and Deployment Center services and utilities. When you **install** or **upgrade** Deployment Center, you can enable HTTPS communication in the *.properties* files.

Obtain SSL/TLS certificates for the Deployment Center server

Deployment Center supports an HTTPS encrypted connection using JKS and PKCS12 certificates. You may use the Deployment Center Vault as the certificate authority to generate certificates locally. Alternatively, you may purchase a certificate from a Certificate Authority (CA) or create a self-signed certificate.

If the KeyStore certificate is not signed by a trusted CA (such as when you create a self-signed certificate), you may provide an associated TrustStore and password.

Set the **Certificate Authority Settings** parameters for Deployment Center to configure how you will obtain certificates.

Enable HTTPS when installing or upgrading Deployment Center

To use HTTPS communication between the Deployment Center server and other machines, set the **useHttpsCommunication** parameter to **true**. To use this setting, you must configure **Certificate Authority Settings** parameters.

By default, **useHttpsCommunication** is set to **false**. Setting it to **true** enables HTTPS and *disables* HTTP communication with Deployment Center.

Deployment Center services and utilities must use HTTPS.

After you enable HTTPS, Deployment Center refuses all HTTP connections. If you receive unexplained web browser errors when accessing the Deployment Center server, determine whether the URL is using HTTP.

Manage secrets using the Deployment Center Vault

Set **Deployment secrets distribution setting** parameters to configure management of passwords and other Deployment Center secrets with the Deployment Center Vault.

The **onlineSecretsDistribution** parameter specifies whether to enable online access to secrets from the Deployment Center Vault. The supported values and modes for this setting are as follows:

true (Online mode) With this setting, deploy scripts retrieve secrets from the Deployment Center Vault during deployment.

> Caution:
>
> If you set **onlineSecretsDistribution** to **true**, you *must* set the **useHttpsCommunication** parameter to **true** to ensure encrypted transmission of secrets from the Deployment Center server.

false (Offline mode) With this setting, secrets are encrypted and distributed in the deployment packages generated by Deployment Center.

HTTPS support for Deployment Center utilities

After you configure HTTPS support, the following utilities use HTTPS:

send_configuration_to_dc

deploy

dc_quick_deploy

Deployment Center architecture

The basic architecture of Deployment Center is comprised of several main parts that communicate with each other.

Jetty web server and the Deployment Center web application

A Jetty web server is automatically installed and configured for Deployment Center, and the installation automatically deploys and runs the Deployment Center web application. No additional installation or configuration is required for the Jetty web server or the Deployment Center web application. Access the web application from a web browser on any machine.

H2 database

The H2 database is also automatically installed and configured for Deployment Center. The database stores information about the Teamcenter environments registered with Deployment Center. No additional installation or configuration is required for the H2 database.

HashiCorp Vault (Deployment Center Vault)

Deployment Center uses HashiCorp Vault to securely store and retrieve deployment secrets. In the installation configuration properties or upgrade configuration properties for Deployment Center, you specify:

- Online or offline retrieval of deployment secrets from the vault
- Service name and port of HashiCorp Vault
- Whether to use the vault as the certificate authority

Repository and the repository service

The repository stores the downloaded software kits and files needed during deployment. Deployment Center uses the repository subdirectories when it registers Teamcenter environments and displays choices for installing and updating software and applications.

You provide the repository directory location during installation of Deployment Center. Be sure that repository location has adequate disk space available to store all the software kits needed for your deployments.

The following repository subdirectories are created automatically during Deployment Center installation or when you perform certain Deployment Center operations:

- **backup**

  Contains a backup of the repository folder when you upgrade the Deployment Center server.

- **dc_contributions**

Contains the Deployment Center files that are needed for Deployment Center to install earlier releases of Teamcenter, Active Workspace, and other supported software.

> Caution:
>
> Do not make changes to this directory.

- **deploy_scripts**

  Contains the deploy scripts Deployment Center generates for an environment. The deploy scripts and related files are generated into a timestamped folder in the respective environment folder.

- **report**

  Contains the generated environment validation report that supports the registration of an existing environment from Teamcenter Environment Manager (TEM) environment.

- **snapshot**

  Contains the Quick Deploy file captured by Deployment Center as an environment snapshot. Environment snapshot files are deleted when the associated Deployment Center environment is deleted.

- **software**

  Contains the Deployment Center-compatible software available to install or update using the current Deployment Center server. These can be Teamcenter 2412 or other versions, a custom Business Modeler IDE software package, or other Deployment Center-compatible software.

  Download Teamcenter software from the **Downloads** page on Support Center. After downloading, expand the **.zip** archive and copy the unzipped folder into the **software** directory.

  Deployment Center automatically scans the *repository-location*/**software** directory and registers any software within this folder for use in Deployment Center.

  Kits available to install are listed in the **Active Media** tab in the **Software Repositories** page.

  You may define multiple software repository locations when you install Deployment Center.

- **system**

  Stores software files that Deployment Center requires to manage deployments.

  > Caution:
  >
  > Do not make changes to this directory.

The repository service is automatically installed when you install Deployment Center. The repository service runs automatically and monitors the repository. The repository service reports the software kits to Deployment Center and populates the list of available software selections.

2. Installing and using Deployment Center

Prepare to install Deployment Center

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 Software Certifications Matrix (**Tc2412PlatformMatrix-*date*.xlsx**), which contains information about system software certified for Deployment Center and Teamcenter, such as operating systems and Java runtime environments (JREs).

   Make sure the required system software is installed on your machine before you install or upgrade Deployment Center and Teamcenter.

3. Install a certified Java runtime environment (JRE) on the Deployment Center machine. Then, set the **JAVA_HOME** and **JRE_HOME** system environment variables to the location of the certified JRE.

Verify system storage space

Siemens Digital Industries Software recommends the following storage space be available on your Deployment Center machine:

- Deployment Center server software: Minimum 2.5 GB.
- Deployment Center software repository: Minimum 12 GB, plus about 56 MB for each generated deployment package.

  The software repository also requires space for downloaded software kits and additional space for any custom software packages. You can store software downloads on the Deployment Center server or you can **configure repository locations** on separate file systems.

Locate the Deployment Center software kit

The Deployment Center software kit is included in the Teamcenter software kit.

1. If you already downloaded the Teamcenter 2412 software kit, skip to step 2. Otherwise, download the Teamcenter 2412 software kit:

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

Products→Teamcenter→Downloads

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

c. Download the Teamcenter 2412 software kit for your platform (**Tc2412_*platform*.zip**). Extract its contents to a local directory.

2. In the Teamcenter 2412 software directory, change to the **install\DeploymentCenter** directory (on Windows systems) or the **install/DeploymentCenter** directory (on Linux systems).

3. Copy the Deployment Center software kit (**deployment_center.zip**) to a directory on your Deployment Center machine.

4. Extract the contents of the Deployment Center software kit on your Deployment Center machine.

Install Deployment Center

Make sure you install Deployment Center using the **Deployment Center software kit** included in the latest Teamcenter software kit you plan to install.

Configure Deployment Center properties, then run the Deployment Center installation script.

Configure Deployment Center installation properties

1. In the directory where you expanded the Deployment Center software kit, change to the **deployment_center** directory. Locate the installation properties file, *install_config.properties*, which specifies parameters for the Deployment Center installation.

2. Create a copy of the installation properties file, for example, **DC_install_config.properties**.

   Open your installation properties file in a text editor to begin setting installation parameters.

3. Set values for the required parameters or accept the default values provided.

   Parameters that are optional are commented out. You may uncomment and set these parameters if you want, but do *not* remove optional parameters from your installation properties file.

   When you are finished setting parameters, save your installation properties file.

Table 2-1. Deployment Center Installation Configuration Parameters

Parameter Group	Parameter	Description
Product Excellence Program Settings	**enableSoftwareAnalytics**	Specifies whether to participate in the Product Excellence Program. Set to **Yes** to participate or **No** to opt out.

Parameter Group	Parameter	Description
		If you do not specify a value, the Deployment Center installation script prompts you to enter a value. For more information, see the **Siemens Digital Industries Software Trust Center**.
Deployment Center Server Settings	**serverDir**	Specifies the full path to the directory for the Deployment Center web server and the database. This directory path must have write access enabled.
	serverPort	Specifies the port where the Deployment Center web server listens for requests. This port must *not* be in use by another service.
	serviceName	Specifies the service name for the Deployment Center web server. The default value is **DC_Service**. Make sure this name does *not* match any existing service name. A duplicate service name causes installation to fail.
	serviceDName	Specifies the display name for the Deployment Center service. The default value is **Deployment Center Service**. This parameter applies to Windows systems only.
Deployment Center Repository Settings	**repoDir**	Specifies the full path to the directory for the primary Deployment Center software repository. The Deployment Center installation script creates a **software** subdirectory in this path. The path must be a local path. UNC paths and mapped drives are *not* supported for the primary repository directory.
	repoPort	Specifies the port where the Deployment Center repository listens for requests from the Deployment Center server. This port must *not* be in use by another service.
	repoServiceName	Specifies the name for the Deployment Center repository service. The default value is **DC_RepoService**. Make sure this name does *not* match any existing service name. A duplicate service name causes installation to fail.
	repoServiceDName	Specifies the display name for the Deployment Center repository service. The default value is **Deployment Center Repository Service**. This parameter applies to Windows systems only.
	repoServicePublisherDName	Specifies the display name for the Deployment Center repository publisher service. The default value is **Deployment Center Repository Publisher Service**. This parameter applies to Windows systems only.
	repoPublisherPort	Specifies the port used by the repository publisher. Do *not* change this value unless it conflicts with a port already in use.
	repoSubscriberPort	Specifies the port used by the repository subscriber. Do *not* change this value unless it conflicts with a port already in use.
Deployment Center User Settings	**user**	Specifies the user name for logging on to Deployment Center. The user views environments, sets up installations, and generates deployment scripts. This user account must have administrative privileges on the Deployment Center machine. The default value is **dcadmin**.

Parameter Group	Parameter	Description
	password	Specifies the password for the Deployment Center user. By default, the password is unencrypted. To use an encrypted password for Deployment Center: a. Create an encrypted password using the **dc_encrypt** utility. b. Set the **useEncryptedPassword** parameter to **true**. c. Enter the encrypted password in the **password** parameter *or* save the encrypted password to a file and specify its location in the **dcpasswordFile** parameter.
	dcpasswordFile (Optional)	Specifies the path to the file that contains the password for the Deployment Center user. If the file contains an encrypted password, be sure to set the **useEncryptedPassword** parameter to **true**.
Extra Software Directory Settings	**extraSoftwareDir** (Optional)	Specifies a comma-separated list of paths to additional software repository directories. You may specify local directory paths or UNC paths. Mapped drives are *not* supported. If you use UNC paths, you must specify values for **osUserNameRepoService** and **osUserPasswordRepoService**.
	osUserNameRepoService (Optional)	Specifies the OS user name for the Deployment Center Repository Service on the current machine. The OS user must exist on the current machine and have access to the UNC paths specified in **extraSoftwareDir**. This parameter is required if you use UNC paths.
	osUserPasswordRepoService (Optional)	Specifies the OS user password for the user specified in **osUserNameRepoService**. This parameter is required if you use UNC paths.
Encrypted Password Settings	**useEncryptedPassword**	Specifies whether to use encrypted passwords for Deployment Center installation. This setting applies to all password parameters in the installation properties file. This parameter is set to **false** by default. Set to **true** to enter encrypted passwords for *all* password parameters.
Script Generation Directory Settings	**osUserNameDCService** (Optional)	Specifies the OS user name to use when generating deployment scripts to UNC paths. This parameter applies only to Windows systems. This parameter is required if you specify any UNC paths in the **scriptsDir***n* parameters.
	osUserPasswordDCService (Optional)	Specifies the OS user password for the user specified in **osUserNameDCService**. This parameter is required if you specify any UNC paths in the **scriptsDir***n* parameters.
	scriptsDir*n* (Optional)	Specifies one or more directories where you want to generate deployment scripts for the dc_quick_deploy utility. If no **scriptsDir** is specified, the default location is the primary Deployment Center repository.

Parameter Group	Parameter	Description
		Append a positive integer to the **scriptsDir** argument for each specification. Then specify the path to the directory. For example, **scriptsDir123=***scripts_path*. You may specify local drives or UNC paths. To use UNC paths, you must run Deployment Center as a Windows service *and* specify values for **osUserNameDCService** and **osUserPasswordDCService**.
HTTPS Communication Settings	**useHttpsCommunication**	Specifies whether to use HTTPS protocol for all communication with Deployment Center. You must have a supported SSL/TLS certificate, an associated KeyStore, and optionally a TrustStore. The default value is **false**. Setting to **true** disables support for HTTP communication. Deployment Center services and utilities must use HTTPS.
	certificateStoreType (Optional)	Specifies the certificate type you use, either **JKS** or **PKCS12**. This parameter is required if you use HTTPS and do not use the Deployment Center Vault as your CA.
	keyStorePath (Optional)	Specifies the path to the KeyStore for your certificate. This parameter is required if you use HTTPS and do not use the Deployment Center Vault as your CA.
	keyStorePassword (Optional)	Specifies the password to the KeyStore specified in **keyStorePath**.
	trustStorePath (Optional)	Specifies the path to the TrustStore for your KeyStore if your certificate is not signed by a known and trusted CA. The TrustStore file is bundled with the Deployment Center utilities to support the KeyStore. This parameter is required if you use a self-signed KeyStore.
	trustStorePassword (Optional)	Specifies the password to the TrustStore specified in **trustStorePath**.
Deployment Secrets Management Settings	**vaultType**	Specifies the vault type for Deployment Center secrets management, that is, the type of HashiCorp vault you use. Two values 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 Deployment Center machine. Deployment Center installs the vault. **ENTERPRISE** — Specifies the enterprise edition of HashiCorp Vault. To use this option, you must install this edition of HashiCorp Vault before you install Deployment Center. You must also create a namespace and supply the URL to the vault and a one-time token to configure the vault.
Deployment Secrets Distribution Settings	**onlineSecretsDistribution**	Specifies whether to enable online access to secrets from the Deployment Center Vault (HashiCorp Vault). Two values are supported: **false (offline mode)** — Secrets are encrypted and distributed with the deployment packages generated by Deployment Center.

Parameter Group	Parameter	Description
		true (online mode) Deploy scripts retrieve secrets from HashiCorp Vault during deployment.
	vaultURL	Specifies the URL to the HashiCorp Enterprise Vault. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultToken	Specifies the one-time token issued by HashiCorp to configure the Enterprise Vault. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultNamespace	Specifies the HashiCorp Enterprise Vault secrets namespace. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultAuthNamespace (Optional)	Specifies the HashiCorp Enterprise Vault authentication namespace. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault *and* your authentication namespace is different from your secrets namespace.
	vaultTenantId (Optional)	Specifies the tenant ID of the HashiCorp Enterprise Vault. The default value is **dc**.
	enterpriseAuthEntityName (Optional)	Specifies entity name of the HashiCorp Enterprise Vault authentication namespace. The default value is **TeamcenterEntity**.
Certificate Authority Settings	**useVaultAsDefaultCA**	Specifies whether to use the Deployment Center Vault as the certificate authority (CA) to generate certificates for HTTPS communication with Deployment Center. This parameter is set to **true** by default. If you use a different CA, set this parameter to **false** and enter values for the remaining parameters in this section.
	vaultServicePort	Specifies the port used by the Deployment Center Vault service.
	vaultServiceClusterPort	Specifies the port used by the Deployment Center Vault service for cluster server requests. Make sure this port is *not* in use by another service.
	vaultServiceName	Specifies the name used by the Deployment Center Vault service. The default value is **DC_Vault_Service**. Make sure this name does *not* match any existing service name. A duplicate service name causes the installation to fail.
	vaultServiceDName	Specifies the display name for the Deployment Center Vault service. The default value is **Deployment Center Vault Service**. This parameter applies to Windows systems only.
	osUserNameVaultService (Optional)	Specifies the OS user name to use to access the Deployment Center Vault service in online mode. (That is, if **onlineSecretsDistribution** is set to **true**.) This setting applies to Windows systems only.
	osUserPasswordVaultService (Optional)	Specifies the OS password for the user specified in **osUserNameVaultService**.

Launch the Deployment Center installation script

1. Open an administrator command prompt and change to the directory where you expanded the Deployment Center software kit. Then, change to the **deployment_center** directory within this directory.

2. Run the Deployment Center installation script, **deployment_center.bat** (on Windows systems) or **deployment_center.sh** (on Linux systems), specifying your updated installation properties file using the **-inputFile** argument. Example:

   - **Windows systems:**

     ```
     deployment_center.bat -install
     -inputFile=DC_install_config.properties
     ```

   - **Linux systems:**

     ```
     deployment_center.sh -install
     -inputFile=DC_install_config.properties
     ```

3. After the installation is complete, the script displays the URL to access Deployment Center:

   https://*machine***:***port***/deploymentcenter**

   Open this URL and enter the user and password you specified in the Deployment Center installation properties file.

4. On Linux systems, Deployment Center creates a post-installation tasks script that installs daemon processes for the Deployment Center server, repository service, publisher service, and vault service. After the installation is complete, run the post-installation tasks script:

   a. As a user with **root** privileges, change to the following directory:

      *DC-server-dir***/rootscripts**

      Replace *DC-server-dir* with the Deployment Center server directory you specified in the **serverDir** parameter in your installation parameters file.

   b. Run the following script:

      ```
      root_servicessetup.sh
      ```

      This script registers daemons and performs other installation actions that require root privileges.

If the installation fails or the installation script displays errors, see the Deployment Center installation log file:

- **Windows systems:**

  *DC-root***deployment_center\webserver\logs**

- **Linux systems:**

  DC-root/**deployment_center/webserver/logs/**

Replace *DC-root* with the Deployment Center installation path you specified in the **serverDir** parameter.

The Deployment Center installation log file also includes the URL to Deployment Center.

> Note:
>
> If you did not install Deployment Center as a service (on Windows systems) or a daemon (on Linux systems) by specifying the **serviceName** parameter in the Deployment Center installation properties, you must **start the Deployment Center web server manually**. You may need to **set the locale** when you do this.

Upgrade Deployment Center

When you upgrade Deployment Center, all its data is preserved, including locations of software repositories and deployed-environment information. Be sure to **check supported upgrade paths** for Deployment Center before you proceed.

During upgrade, you may make changes to your existing installation, such as the following:

- Migrate to using the **HTTPS protocol** for communication with the Deployment Center server.
- Add repository locations.
- Update where you want to generate deployment scripts.

Make sure you upgrade Deployment Center using the **Deployment Center software kit** included in the latest Teamcenter software kit you plan to install.

Configure Deployment Center upgrade properties, and then run the Deployment Center upgrade script.

Configure Deployment Center upgrade properties

1. **Verify system software requirements** for Deployment Center, and **locate the Deployment Center software kit**.

2. In the directory where you expanded the Deployment Center software kit, change to the **deployment_center** directory. Locate the upgrade properties file, named *upgrade_config.properties*, which specifies parameters for the Deployment Center upgrade.

3. Create a copy of the installation properties file, for example, **DC_upgrade_config.properties**.

   Open your installation properties file in a text editor to begin setting installation parameters.

4. Set values for the required parameters, or accept the default values provided.

   Parameters that are optional are commented out. You may uncomment and set these parameters if you want, but do *not* remove optional parameters from your upgrade properties file.

   When you are finished setting parameters, save your upgrade properties file.

Table 2-2. Deployment Center Upgrade Configuration Parameters

Parameter Group	Parameter	Description
Deployment Center Server Settings	**serverDir**	Specifies the full path to the directory for the Deployment Center web server and the database. This directory path must have write access enabled.
Deployment Center Repository Settings	**repoPublisherPort**	Specifies the port used by the repository publisher. Do *not* change this value unless it conflicts with a port already in use.
	repoSubscriberPort	Specifies the port used by the repository subscriber. Do *not* change this value unless it conflicts with a port already in use.
Deployment Center User Settings	**user**	Specifies the user name for logging on to Deployment Center. The user views environments, sets up installations, and generates deployment scripts. This user account must have administrative privileges on the Deployment Center machine.
	password	Specifies the password for the Deployment Center user. By default, the password is unencrypted. To use an encrypted password for Deployment Center: a. Create an encrypted password using the **dc_encrypt** utility. b. Set the **useEncryptedPassword** parameter to **true**. c. Enter the encrypted password in the **password** parameter, *or* save the encrypted password to a file and specify its location in the **dcpasswordFile** parameter.
	serviceDName	Specifies the display name for the Deployment Center service. The default value is **Deployment Center Service**. This parameter applies to Windows systems only.
	repoServiceDName	Specifies the display name for the Deployment Center repository service. The default value is **Deployment Center Repository Service**. This parameter applies to Windows systems only.

Parameter Group	Parameter	Description
	repoServicePublisherDName	Specifies the display name for the Deployment Center repository publisher service. The default value is **Deployment Center Repository Publisher Service**. This parameter applies to Windows systems only.
Extra Software Directory Settings	**extraSoftwareDir** (Optional)	Specifies a comma-separated list of paths to additional software repository directories. You may specify local directory paths or UNC paths. Mapped drives are *not* supported. If you use UNC paths, you must specify values for **osUserNameRepoService** and **osUserPasswordRepoService**.
	osUserNameRepoService (Optional)	Specifies the OS user name for the Deployment Center Repository Service on the current machine. The OS user must exist on the current machine and have access to the UNC paths specified in **extraSoftwareDir**. This parameter is required if you use UNC paths.
	osUserPasswordRepoService (Optional)	Specifies the operating system user password for the user specified in **osUserNameRepoService**. This parameter is required if you use UNC paths.
Encrypted Password Settings	**useEncryptedPassword**	Specifies whether to use encrypted passwords for Deployment Center installation. This setting applies to all password parameters in the installation properties file. This parameter is set to **false** by default. Set to **true** to enter encrypted passwords for *all* password parameters.
Script Generation Directory Settings	**osUserNameDCService** (Optional)	Specifies the OS user name to use when generating deployment scripts to UNC paths. This parameter applies only to Windows systems. This parameter is required if you specify any UNC paths in the **scriptsDir***n* parameters.
	osUserPasswordDCService (Optional)	Specifies the OS user password for the user specified in **osUserNameDCService**. This parameter is required if you specify any UNC paths in the **scriptsDir***n* parameters.
	scriptsDir*n* (Optional)	Specifies one or more directories where you want to generate deployment scripts for **the dc_quick_deploy utility**. If no **scriptsDir** is specified, the default location is the primary Deployment Center repository. Append a positive integer to the **scriptsDir** argument for each specification. Then specify the path to the directory, for example, **scriptsDir123=***scripts_path*. You may specify local drives or UNC paths. To use UNC paths, you must run Deployment Center as a Windows service *and* specify values for **osUserNameDCService** and **osUserPasswordDCService**.
HTTPS Communication Settings	**useHttpsCommunication**	Specifies whether to use HTTPS protocol for all communication with the Deployment Center. You must have a supported SSL/TLS certificate, an associated KeyStore, and optionally a TrustStore. Setting this parameter to **true** disables support for

Parameter Group	Parameter	Description
		HTTP communication. Deployment Center services and utilities must use HTTPS.
Deployment Secrets Management Settings	**vaultType**	Specifies the vault type for Deployment Center secrets management, that is, the type of HashiCorp vault you use. Two values 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 Deployment Center machine. Deployment Center installs the vault. **ENTERPRISE** — Specifies the enterprise edition of HashiCorp Vault. To use this option, you must install this edition of HashiCorp Vault before you install Deployment Center. You must also create a namespace and supply the URL to the vault and a one-time token to configure the vault.
Deployment Secrets Distribution Settings	**onlineSecretsDistribution**	Specifies whether to enable online access to secrets from the Deployment Center Vault (HashiCorp Vault). Two values are supported: **false (offline mode)** — Secrets are encrypted and distributed with the deployment packages generated by Deployment Center. **true (online mode)** — Deploy scripts retrieve secrets from HashiCorp Vault during deployment.
	vaultURL	Specifies the URL to the HashiCorp Enterprise Vault. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultToken	Specifies the one-time token issued by HashiCorp to configure the Enterprise Vault. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultNamespace	Specifies the HashiCorp Enterprise Vault secrets namespace. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault (**vaultType**=**ENTERPRISE**).
	vaultAuthNamespace (Optional)	Specifies the HashiCorp Enterprise Vault authentication namespace. Enable (uncomment) and set this parameter if you use HashiCorp Enterprise Vault *and* your authentication namespace is different from your secrets namespace.
	vaultTenantId (Optional)	Specifies the tenant ID of the HashiCorp Enterprise Vault. The default value is **dc**.
	enterpriseAuthEntityName (Optional)	Specifies entity name of the HashiCorp Enterprise Vault authentication namespace. The default value is **TeamcenterEntity**.
Certificate Authority Settings	**useVaultAsDefaultCA**	Specifies whether to use the Deployment Center Vault as the certificate authority (CA) to generate certificates for HTTPS communication with Deployment Center. This parameter is set

Parameter Group	Parameter	Description
		to **true** by default. If you use a different CA, set this parameter to **false** and enter values for the remaining parameters in this section.
	certificateStoreType (Optional)	Specifies the certificate type you use, either **JKS** or **PKCS12**. This parameter is required if you use HTTPS and do not use the Deployment Center Vault as your CA.
	keyStorePath (Optional)	Specifies the path to the KeyStore for your certificate. This parameter is required if you use HTTPS and do not use the Deployment Center Vault as your CA.
	keyStorePassword (Optional)	Specifies the password to the KeyStore specified in **keyStorePath**.
	trustStorePath (Optional)	Specifies the path to the TrustStore for your KeyStore if your certificate is not signed by a known and trusted CA. The TrustStore file is bundled with the Deployment Center utilities to support the KeyStore. This parameter is required if you use a self-signed KeyStore.
	trustStorePassword (Optional)	Specifies the password to the TrustStore specified in **trustStorePath**.
Deployment Secrets Distribution Settings	**onlineSecretsDistribution**	Specifies whether to enable online access to secrets from the Deployment Center Vault. If set to **false** (offline mode), secrets are encrypted and distributed with the deployment packages generated by Deployment Center. If set to **true** (online mode), deploy scripts retrieve secrets from the Deployment Center Vault during deployment.
	vaultServicePort	Specifies the port used by the Deployment Center Vault service.
	vaultServiceClusterPort	Specifies the port used by the Deployment Center Vault service for cluster server requests. Make sure this port is *not* in use by another service.
	vaultServiceName	Specifies the name used by the Deployment Center Vault service. The default value is **DC_Vault_Service**. Make sure this name does *not* match any existing service name. A duplicate service name causes installation to fail.
	vaultServiceDName	Specifies the display name for the Deployment Center Vault service. This parameter applies to Windows systems only.
	osUserNameVaultService (Optional)	Specifies the OS user name to use to access the Deployment Center Vault service in online mode (that is, if **onlineSecretsDistribution** is set to **true**). This setting applies to Windows systems only.
	osUserPasswordVaultService (Optional)	Specifies the OS password for the user specified in **osUserNameVaultService**.

Caution:

Do *not* remove the repository subscriber port (**repoSubscriberPort**) or repository publisher port (**repoPublisherPort**) parameters. These parameters are for internal use by the

repository. If the port specification conflicts with a port already in use, you may change the port to an unused port.

Launch the Deployment Center upgrade script

1. Open an administrator command prompt and change to the directory where you expanded the Deployment Center software kit. Then, change to the **deployment_center** directory within this directory.

2. Run the Deployment Center installation script, **deployment_center.bat** (on Windows systems) or **deployment_center.sh** (on Linux systems), specifying your updated upgrade properties file using the **-inputFile** argument. Example:

   - **Windows systems:**

     ```
     deployment_center.bat -upgrade
     -inputFile=DC_upgrade_config.properties
     ```

   - **Linux systems:**

     ```
     deployment_center.sh -upgrade
     -inputFile=DC_upgrade_config.properties
     ```

3. After the upgrade is complete, the script displays the URL to access Deployment Center:

   https://*machine***:***port***/deploymentcenter**

   Open this URL and enter the user and password you specified in the Deployment Center upgrade properties file.

If the installation fails or the installation script displays errors, see the Deployment Center log file:

- **Windows systems:**

  *DC-root***\deployment_center\webserver\logs**

- **Linux systems:**

  *DC-root***/deployment_center/webserver/logs/**

Replace *DC-root* with the Deployment Center installation path you specified in the **serverDir** parameter.

The Deployment Center installation log file also includes the URL to Deployment Center.

Note:

If you did not install Deployment Center as a service (on Windows systems) or a daemon (on Linux systems) by specifying the **serviceName** parameter in the Deployment Center installation properties, you must **start Deployment Center manually**.

You may need to **set the locale on the Deployment Center URL**.

If you want to change the Deployment Center administrator password, use the **deployment_center.bat -maintenance** command.

If you encounter problems during Deployment Center upgrade, see **Troubleshoot Deployment Center installation or upgrade**.

Maintain Deployment Center

You can perform Deployment Center maintenance actions using the Deployment Center utility in maintenance mode.

To use maintenance mode, run the **deployment_center.bat|sh** utility with the **-maintenance** option. For example:

deployment_center.bat -maintenance

Run the Deployment Center utility in an administrator command prompt. (On Windows, you must be logged on using an account with administrator privileges.)

After the **deployment_center** command and the **-maintenance** option, add the required arguments to the command line, and then add optional arguments or options to perform maintenance actions on the Deployment Center server:

Argument/Option Type	Argument/Option	Description
Required Arguments	**-serverDir** (required)	Specifies the full path to the directory of the Deployment Center installation. This directory path must have write access enabled.
	user (required)	Specify the user name for the Deployment Center administrator.
	password (required)	Specify the password for the Deployment Center user. If **useEncryptedPassword** was set to **True** when the password was entered previously, enter the encrypted password.

Argument/Option Type	Argument/Option	Description
Password Change	**-changePassword**	Specify this option to change the Deployment Center password.
	-newPassword	Specify the new password. You may specify a clear text password or an encrypted password. The current password will be authenticated to make the change.
	-useEncryptedPassword	Specifies whether to use an encrypted password. Set to **true** if you are providing an encrypted password.
Software Repository Settings	**-addSoftwareDir**	Specifies additional paths to add to the **software** directories for the repository. These paths must be separated by commas (**,**). You may specify local file system paths or UNC paths. Mapped drives are not supported. For network locations, you must additionally specify the user name and password: **-osUserNameRepoService** — Specify the OS user name for the Deployment Center Repository Service on the current machine. The OS user must exist on the current machine and should have access to the UNC paths specified in the additional software directories. **-osUserPasswordRepoService** — Specify the OS user password for the Deployment Center Repository Service on the current machine. To reiterate, this OS user must exist on the current machine and should have access to the UNC paths specified in the additional software directories.
	-removeSoftwareDir	Specifies directories to remove from Deployment Center repository scanning. These must be existing paths to **software** directories that are scanned by the repository. Separate multiple paths by commas (**,**). This action does not delete **software** directories, and it does not remove scanned software from the repository display. However, the repository no

Argument/Option Type	Argument/Option	Description
		longer scans the directory for additional software. Software in the directory is not removed from the system. Do not remove a directory specified as the primary repository, by the **-repoDir argument during installation**.
Software Repository Service Settings Note: When you run the utility with these arguments, Deployment Center uninstalls and deletes the existing service, then reinstalls it with the new properties and restarts the services.	**-changeRepoDir**	Changes the repository directory to the specified path. For example: **-changeRepoDir=C:\dc_repo1**
	-changeRepoPort	Changes the port used by the repository service to the specified value. For example: **-changeRepoPort=8082**
	-changeRepoServiceName	Changes the service name of the repository service to the specified value. For example: **-changeServiceName=DC_Repo_Service**
	-changeRepoPublisherPort	Changes the port used by the Deployment Center repository publisher service to the specified value. For example: **-changeRepoPublisherPort=8083**
	changeRepoSubscriberPort	Changes the port used by the repository subscriber to the specified value. **-changeRepoSubscriberPort=8084**
Deployment Script Directory Settings	**-addScriptsDir**	Specifies one or more directories where you want to generate deployment scripts for **the dc_quick_deploy utility**. Append an integer to each **scriptsDir***n* argument followed by a colon (**:**) and the associated path, for example, **scriptsDir123:***scripts_path*. Separate multiple identifiers by commas (**,**). You may specify local drives or UNC paths. For network locations, specify the user name and password. For UNC paths, Deployment Center must be running as a Windows service.

Argument/Option Type	Argument/Option	Description
		If you specify an existing **scriptsDir***n*, the previous value will be overwritten by the new value. For network locations, you must additionally specify the user name and password: **-osUserNameDCService** Specify the user name for the UNC or mapped drive. **-osUserPasswordDCService** Specify the user password for the UNC or mapped drive.
	-removeScriptsDir	Specifies one or more locations you want to remove from deployment script generation. Specify the **scriptsDir***n* argument that defines the location, for example, **scriptsDir123**. Separate multiple identifiers with commas (**,**).
Deployment Center Service Settings When you run the utility with these arguments, Deployment Center uninstalls and deletes the existing service, then reinstalls it with the new properties and restarts the services.	**-changeServiceName**	Changes the service name of the Deployment Center service to the specified value. For example: **-changeServiceName=DC_Service**
	-changeServerPort	Changes the port used the Deployment Center service to the specified value. For example: **-changeServerPort=8081**
Certificate Authority Settings	**-updateCertificates**	Updates the Deployment Center certificates. If your default certificate authority (CA) is the Deployment Center Vault, the vault regenerates certificates. If you use a certificate provide by a third-party certificate authority (CA), you must include the following additional arguments: **-keyStorePath** Specifies the path to the KeyStore for your certificate. **-trustStorePath** Specifies the path to the TrustStore for your KeyStore if your certificate is not signed by a known and trusted CA. The

Argument/Option Type	Argument/Option	Description
		TrustStore file is bundled with the Deployment Center utilities to support the KeyStore. This parameter is required if you use a self-signed KeyStore. **-trustStorePassword** Specifies the password to the TrustStore specified in trustStorePath. This required only if -trustStorePath is specified. **-certificateStoreType** Specifies the certificate type you use, either **JKS** or **PKCS12**. This parameter is required if you use HTTPS and do not use the Deployment Center Vault as your CA.
	-changeUseVaultAsDefaultCA	Specifies whether to use the Deployment Center Vault as the certificate authority (CA) to generate certificates for HTTPS communication with Deployment Center. You specified this setting during Deployment Center installation, but you can change it in maintenance mode. Set to one of the following values: **true** Sets the Deployment Center Vault as the default certificate authority that generates and signs certificates. No additional arguments are required. **false** Sets the certificate authority to a Vendor CA, if Deployment Center was initially configured as default certificate authority. If you use this setting, set the additional arguments described above for the **updateCertificates** option: **-keyStorePath** **-trustStorePath** **-trustStorePassword** **-certificateStoreType**
	-changeVaultServiceName	Changes the service name of the Deployment Center Vault service to the specified value.

Argument/Option Type	Argument/Option	Description
	-changeVaultServicePort	Changes the port used by the Deployment Center Vault service to the specified value.
Product Excellence Program	**-enableSoftwareAnalytics**	Specifies whether to participate in the Siemens Product Excellence Program. You specified this setting **during Deployment Center installation**, but you can change it in maintenance mode. Set **-enableSoftwareAnalytics=yes** to enable participation in the program, or set it to **no** to decline participation. The Product Excellence Program anonymously shares information with Siemens Digital Industries Software about how you deploy Deployment Center software, to help Siemens improve future releases of Deployment Center. Data collection occurs in the background as software is used and does not affect performance or functionality. For more information about the Product Excellence Program, see the Siemens **Trust Center**.

On Windows systems, when specifying paths in maintenance mode arguments, you may specify local drives or UNC paths. When you specify a path, use double backslashes (****) to make the path valid for Java.

Caution:

When you add or remove directories, be sure that no one is using the Deployment Center for any activity during maintenance. Deployment Center services are stopped and restarted during the update.

Example:

Add and remove directories for the repository list and the generated scripts list:

```
deployment_center.bat -maintenance -serverDir=D:\dc\win\4
   -user=dcadmin1 -password=dcadmin1
-addSoftwareDir=D:\Foundation_software,D:\software2
   -removeSoftwareDir=X:\Foundation_software,X:\software2
   -addScriptsDir=scriptsDir10:D:\abc1,scriptsDir11:D:\abc2
   -removeScriptsDir=scriptsDir3,scriptsDir4
```

Uninstall Deployment Center

To remove Deployment Center from your system, perform the following steps to uninstall Deployment Center.

Remove Deployment Center services, the software repository, and Deployment Center files.

1. Open an administrator command prompt on the server where Deployment Center is installed.

2. Stop all Deployment Center services.

   On Windows systems, use the **net stop** command:

   net stop *service-name*

   Enter this command for all Deployment Center services you installed as Windows services:

Displayed name	Service name
Deployment Center Repository Publisher Service	**DC_RepoService_Publisher**
Deployment Center Repository Service	**DC_RepoService**
Deployment Center Service	**DC_Service**
Deployment Center Vault Service	**DC_Vault_Service**

   Example:

```
net stop DC_RepoService_Publisher
net stop DC_RepoService
net stop DC_Service
net stop DC_Vault_Service
```

   This example uses default service names. If you changed any of these service names in the installation parameters when you **installed Deployment Center**, use the service names you specified to stop the services.

   To find a service's service name using the Windows **Services** panel, right-click its displayed name and choose **Properties**.

3. Delete Deployment Center services.

   On Windows systems, use the **sc delete** command:

   sc delete *service-name*

   Example:

```
sc delete DC_RepoService_Publisher
sc delete DC_RepoService
sc delete DC_Service
sc delete DC_Vault_Service
```

If you used non-default service names when you **installed Deployment Center**, use the service names you specified to delete the services.

4. Close the command prompt.

5. Delete the software repository directory. You specified this location in the **repoDir** parameter when you **installed Deployment Center**.

   If you specified additional software directories in the **extraSoftwareDir** parameter, delete those directories also.

6. Delete the entire directory where you installed Deployment Center.

Troubleshoot Deployment Center installation or upgrade

If you encounter problems installing or upgrading Deployment Center, access the log files from the location where you extracted the Deployment Center installation ZIP file. The logs are in *webserver\logs*.

Running the installation script

Make sure you run the Deployment Center installation script from a location with no spaces or quotation marks in the path. For example, this path contains a space:

D:\DC Kits\deployment_center\deployment_center

The space in the path (*DC Kits*) may cause the following error:

```
Could not find or load main class Kits\deployment_center
   \deployment_center\deployment_center\jar\com.dc.jrechecker.jar
   'JAVA_HOME' is set to an unsupported version of Java.
Remove the space in the path.
```

Remove the space in the path.

Repository service name

ff the default **RepositoryService** service name or a service name specified by **repoServiceName** matches an existing service name, the installation returns an error. Check the service names on the server for conflicts. You can either change a conflicting service name or specify a different repository service name using **repoServiceName**.

Service names differ between new and upgraded installations

On Windows systems, if you upgraded an installation of Deployment Center and also performed a new installation of Deployment Center 2412, you may notice the displayed names of the Deployment Center

services differ between the new and upgraded installations. This does not cause problems and the services will operate normally.

Service	Displayed name prior to version 14.1	Displayed name in versions 14.1 and later
Deployment Center Service	**DC_Service**	**Deployment Center Service**
Deployment Center Repository Service	**DC_RepoService**	**Deployment Center Repository Service**

Locale support for the Deployment Center URL

You may need to set the browser locale for Deployment Center as it does not read the web browser locale language. Set the locale by appending the locale code to the end of the Deployment Center URL.

Example:

http://dc_server:8080/deploymentcenter/?locale=de_DE

Supported language codes:

cs_CZ, de_DE, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pl_PL, pt_BR, ru_RU, zh_CN, zh_TW

Start Deployment Center

Before you access the Deployment Center web application from a web browser, start the Deployment Center web server. You can choose either of the following ways:

- Automatically start the server as a service or daemon

  On Windows systems, if you specified the **serviceName** parameter during installation, the Deployment Center web server is installed as a service and starts automatically.

  On Linux systems, the **root_servicessetup.sh** script installs the Deployment Center service as a daemon.

- Start the server manually

  If you did not specify the service arguments, start the Deployment Center service from its startup script. Navigate to the server directory specified by the **serverDir** parameter when you installed Deployment Center and run the **startdc.bat** script.

If you experience a problem starting Deployment Center, see **Troubleshoot Deployment Center operations**.

Log on to Deployment Center

1. Enter the Deployment Center URL in the web browser. The form of the URL is:

   `http://`*host*`:`*serverPort*`/deploymentcenter`

   host is the server where Deployment Center is installed.

   serverPort is the port number specified by the **serverPort** parameter for the **Deployment Center installation script**.

   > Tip:
   >
   > The URL is displayed in the command window after Deployment Center installation is complete.
   >
   > The URL is also specified in the installation log file.

2. Enter the user name and password as specified in the **user** and **password** parameters for the Deployment Center installation script.

   The Deployment Center home page appears.

If you experience a problem in logging on to Deployment Center, see **Troubleshoot Deployment Center operations**.

Troubleshoot Deployment Center operations

You can refer to the log files if you have difficulty with the Deployment Center server operations. The log files are located in *webserver\logs* and may contain information about:

- Inability to log on to Deployment Center.
- Inability to add software, applications, or components to a Deployment Center environment.
- Failure to create a deployment script.
- Deployment Center operation failures such as internal server error, display problems, or missing configuration files.

If you **configure HTTPS for Deployment Center**, it refuses all HTTP connections. If you get an unexplained web browser error when accessing the Deployment Center server, check whether the URL is using **HTTP** instead of **HTTPS**.

If you have a problem with **the browser locale for Deployment Center** as it does not read the browser locale language, you can append the locale code to the end of the Deployment Center URL.

Deployment Center server backup and recovery

To prevent loss of Deployment Center data in the event of a failure, back up Deployment Center software regularly. File backups can be used to recover from configuration errors that cannot otherwise be changed in the system, and can also be used to recover data from a database corruption.

Backup and recovery procedures rely on the **Deployment Center installation process** as and require Deployment Center installation paths and parameters. In the following procedures, *DC-install-dir* refers to the installation path to Deployment Center.

Back up the Deployment Center system

> Note:
>
> As a good practice, perform a backup when you add software to Deployment Center.

1. Stop the Deployment Center server.
2. If you set up a repository service, stop the service. To locate the service, look for the service name or display name which you specified in the **serviceName** and **serviceDName** parameters when you installed Deployment Center.
3. Copy these database files to a safe location:

DC-install-dir\serverDir\db\deploy_center.mv.db

DC-install-dir\serverDir\db\deploy_center.trace.db

4. Record the list of software packages you downloaded to the repository. The repository directory is set to the location specified by **repoDir** during Deployment Center installation, for example:

   DC-install-dir`\repository\software`

   > Note:
   >
   > If you already recorded the list of software packages and there have been no changes to the repository since the last backup, you can skip this step.

5. Restart the repository service and start Deployment Center.

   If you encounter a problem, stop Deployment Center and the repository service and replace the database files from your backup. If that does not fix the problem when you restart Deployment Center, download and replace the software packages in the repository.

Reinstall Deployment Center

If your recovery is not successful, you may need to reinstall Deployment Center.

1. Stop Deployment Center and delete the repository service.

2. Move or rename the existing Deployment Center installation, in case you need to access the software directories to repopulate the repository.

3. Reinstall Deployment Center in the same location, using the same paths and ports as the original installation. To restore the database files, the repository path specified by **repoDir** must be identical to the previous installation.

4. Update the repository with the same software packages that you retrieved from the previous backup. If the repository software packages are available from the installation that you moved or renamed, copy those files.

   Wait for Deployment Center to update and recognize the software.

5. Stop the Deployment Center server.

6. If you set up a repository service, stop the service.

7. Copy the backup database files to your current Deployment Center installation:

   DC-install-dir\serverDir\db\deploy_center.h2.db

DC-install-dir\serverDir\db\deploy_center.trace.db

8. Restart the repository service and other services and start Deployment Center.

   Deployment Center is restored to the state it was in from the last backup.

3. Managing repositories

Manage repositories

Deployment Center stores the software kits for your Teamcenter environment in a repository. The repository helps you determine whether you have everything you need for an installation or update. To access the repositories, log on to Deployment Center and click **SOFTWARE REPOSITORIES**.

Specifying repository locations

Each Deployment Center repository contains a *software* directory that stores your unzipped software kits. The primary repository is specified by the **repoDir parameter** during Deployment Center installation.

You may have Teamcenter software in more than one location. You can add these locations using the **extraSoftwareDir** parameter. Specify the paths for Deployment Center during **installation**, **upgrade**, or **maintenance**. Each specified path presumes there is a *software* subdirectory where unzipped software kits are stored.

The primary repository location (**repoDir**) must be a local directory path. UNC paths and mapped drives are not supported.

Additional repository locations (**extraSoftwareDir**) can be local file system paths or UNC paths. Mapped drives are not supported.

Manage software kits

Deployment Center scans for software in unzipped subdirectories of these registered repository *software* directories. Deployment Center reads directories nested one level under *software*. These subdirectories must contain unzipped, valid software directories.

Deployment Center does not scan further than the first level subdirectories under *software*. Software nested under another software folder is not found. Software that is placed in another location on the server is not scanned.

Review repository contents

The **Active Media** tab displays information about each software kit in the **Software Media** table. Deployment Center refers to the base Teamcenter software as Foundation software.

Software kit information includes the name, version, release type, repository folder location, and when it was last scanned. You can see whether it's in use by a Teamcenter environment and whether it has a dependency.

Dependent software must be present to successfully **choose software to install or update**. If you are missing dependent software, Deployment Center displays a message explaining what you need.

If the repository scanner detects an invalid file, the repository displays the file with a message. Invalid files can be a Word or Excel document, a directory containing software not supported by Deployment Center, or a software ZIP file that is not unzipped.

The repository service scanner updates the repository list periodically, but the refresh rate may be affected by factors such as server location and network performance.

Add or remove repository locations

You can add and remove software locations in the repository at any time using **deployment_center.bat** in **maintenance mode**.

Review software configuration reports

You can review configuration information for any software listed in the repository, and the applications and components the given version supports. The report contains information about products, applications, components, and configurable properties you may need to set for a Teamcenter environment using **the quick deploy process**.

Choose a software kit from the **Active Media** tab and click **Generate Software Configuration Report**  . The HTML results provide comprehensive information about software dependencies and IDs, application internal names and dependencies, and component and property internal names and configuration values. This information can help you create or update the quick deploy **XML configuration file**.

Add software kits into the repository

1. Download the software kits for the software versions that you want to deploy in your Teamcenter environment. Be sure to get the current full release and any updates you want to apply to your Teamcenter environment.

2. Unzip the software kits and copy the unzipped directories to the *software* subdirectory in one of your registered 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 it is correct and complete for your planned deployment. Check whether missing dependencies are noted. If so, retrieve the missing software and check again. The list may take a little while to update.

If you experience a problem in adding software to the Deployment Center repository, you can try to **troubleshoot the 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.

Update repository properties

The **Overview** page provides helpful information about the server where the primary repository is located. The system information includes disk space, which you can monitor when you put software kits in the repository. The available and used disk space values are updated when you add or change software kits in the primary repository.

You can edit some of the repository properties.

1. Click **Start Edit**.

2. You can change the following information:

   - **Repository Name**

     Enter or update a name for the repository.

   - **Location**

Enter or update a location for the repository server. You can use any convention you wish, such as geographical location, the name of a facility, or another value that helps you define the location.

- **Comments**

  Enter or update information that might be helpful to an administrator.

3. Click **Save Edits** .

   To cancel your changes, click **Cancel Edits** .

Maintaining repositories

Use **the maintenance procedure for deployment_center.bat** to add, remove, or update the repository software locations being used by Deployment Center.

Troubleshoot the repository service

Software is not listed in the Software Repositories page

If software placed you placed into the repository does not appear on the **Software Repositories** page, this could indicate a problem with software files or that an error occurred when Deployment Center scanned the repository. View the software repository logs for information about problems encountered during repository scanning. The repository log files are located on the Deployment Center server in *webserver\repotool\logs*.

Out of memory error when scanning large repositories

If your software repository is large, for example, 40 or more items, you could encounter an error similar to the following when Deployment Center scans the repository:

```
java.lang.OutOfMemoryError: Java heap space.
```

The scan process may terminate and the scan results may be left incomplete.

To resolve this issue, you can increase Java memory allocation by editing the following registry key in the Windows Registry Editor:

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Apache Software Foundation\Procrun 2.0\DC_RepoService\Parameters\Java

Replace the old data values with the new data values shown:

Name	Data Value (Old)	Data Value (New)
JvmMs	128	512
JvmMx	512	1024

4. Managing Teamcenter environments

Manage environments

The **Environments** page lists all environments being tracked by Deployment Center. You can view, create, and delete environments. To access the list of environments, log on to Deployment Center and click **ENVIRONMENTS**.

Deploy software in an environment

The **Deploy Software** page displays the selected software used by registered environments. You choose the software necessary to proceed with deploying updates. If an environment is new, the selected software list may be empty. This is where you begin the software deployment process.

The **Status** column shows the installation status of the selected software:

Icon	Status	Description
	Pending Install	The environment is being configured. Deployment scripts may be generated but have not been run.
	In Review	An environment scanned from TEM is in the validation process.
	Install in Progress	Deployment scripts are running, sending current status to Deployment Center.
	Installed	Deployment scripts have run successfully.
	Failed to Install	Failures occurred while running deployment scripts
	Pending Update	Updates to installed software are being configured.

View an environment's properties

The **Overview** page displays the environment's properties, such as the architecture, site, software version, applications, and components installed in the environment. Click an environment to learn more about it.

- Review the **Properties** for the environment.

  If an environment is new, the **Overview** page displays the information used to create it.

  You can edit some environment properties.

- Review the **Software**, **Applications Installed**, and **Components** for an existing environment. You can click a selection to display more information in the right pane.

If an environment is new, **Software**, **Applications Installed**, and **Components** may not be populated, so no Teamcenter environment information may be displayed.

Capturing a snapshot of a Teamcenter environment

Deployment Center can create a *snapshot* of an environment, which records the applications, components, parameter settings, deployment status, and other properties in the form of a Quick Deploy XML file. Snapshots can be used to **capture an environment in a current state** and to roll an environment back to a previous state if necessary.

Restoring an environment to a prior state using a snapshot

If you have captured snapshots of an environment, you can **restore an environment to the state at which the snapshot was captured**. Components that were partially configured in the snapshot are restored to their partially configured states.

Existing deployment scripts cannot be used after restoring a snapshot. You must generate new deploy scripts in the **Deploy** tab.

Exporting environments

You can use an existing environment as a **starting point to create a new environment**. You can export an **XML configuration file** that contains the selected environment's configuration. Edit this configuration file and then reuse it as the input XML file for the **dc_quick_deploy utility**.

On the **Environments** page, choose an environment from the list, and run **Export Environment**. Select whether you want the output XML file to contain all component configuration properties or just the required component properties.

Information about the software is available from the **Software Configuration Report**. The report may be helpful in making updates to the XML configuration file.

Verify software, applications, and components

On the **Environments** page, you can review the software, applications, and components for each environment.

Software	You can verify the status of software for the selected environment. The list includes installed and pending software. You can select the software package to see additional software details.
Applications Installed	You can verify the status of applications for the selected environment. The list includes installed and pending applications. You can select an application package to see additional application details.

Applications are associated with their installed software, such as Search for Active Workspace.

Components You can verify the status of components for the selected environment. The list includes installed and pending components. You can select the component package to see additional information about component settings. Components are associated with their applications, such as Indexing Engine and Indexer for Search.

The component information in the right pane offers two views of the information. In the upper right corner, you can choose the view:

Show only required parameters

Required parameters view displays only required parameter information. Click to expand the view to display both required and optional parameters.

Show all parameters

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

Edit environment properties

On the **Environments** page, you can edit some environment properties. Choose **Overview** to display the properties of the selected environment.

1. Click **Start Edit** to display the editable fields.

2. You can change the following information:

   - **Environment Name**

     Displays the name provided during setup for the environment. You can update the name for the environment.

     > Note:
     >
     > An environment name cannot contain spaces.

   - **Environment Type**

     Displays the type of the environment. The available types are **Integration**, **Development**, **Production**, **Test**, and **Training**. The type is set to **Production** by default when the environment is registered, but you can select another type.

- **Config ID**

  Displays the configuration ID for an environment scanned from TEM.

- **Site ID**

  Displays an auto-generated site ID for the environment. When you copy an environment, Deployment Center generates a different site ID.

- **Location**

  Displays the location of the environment. You can enter or update the geographical location for the environment, such as a city, the name of a facility, or another value that helps you define the location of the environment.

- **Comments**

  Displays additional information entered by the administrator. You can enter or update information about the environment.

3. Click **Save Edits**, or click **Cancel Edits** to discard your changes.

Create environments

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 and validate environments

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**

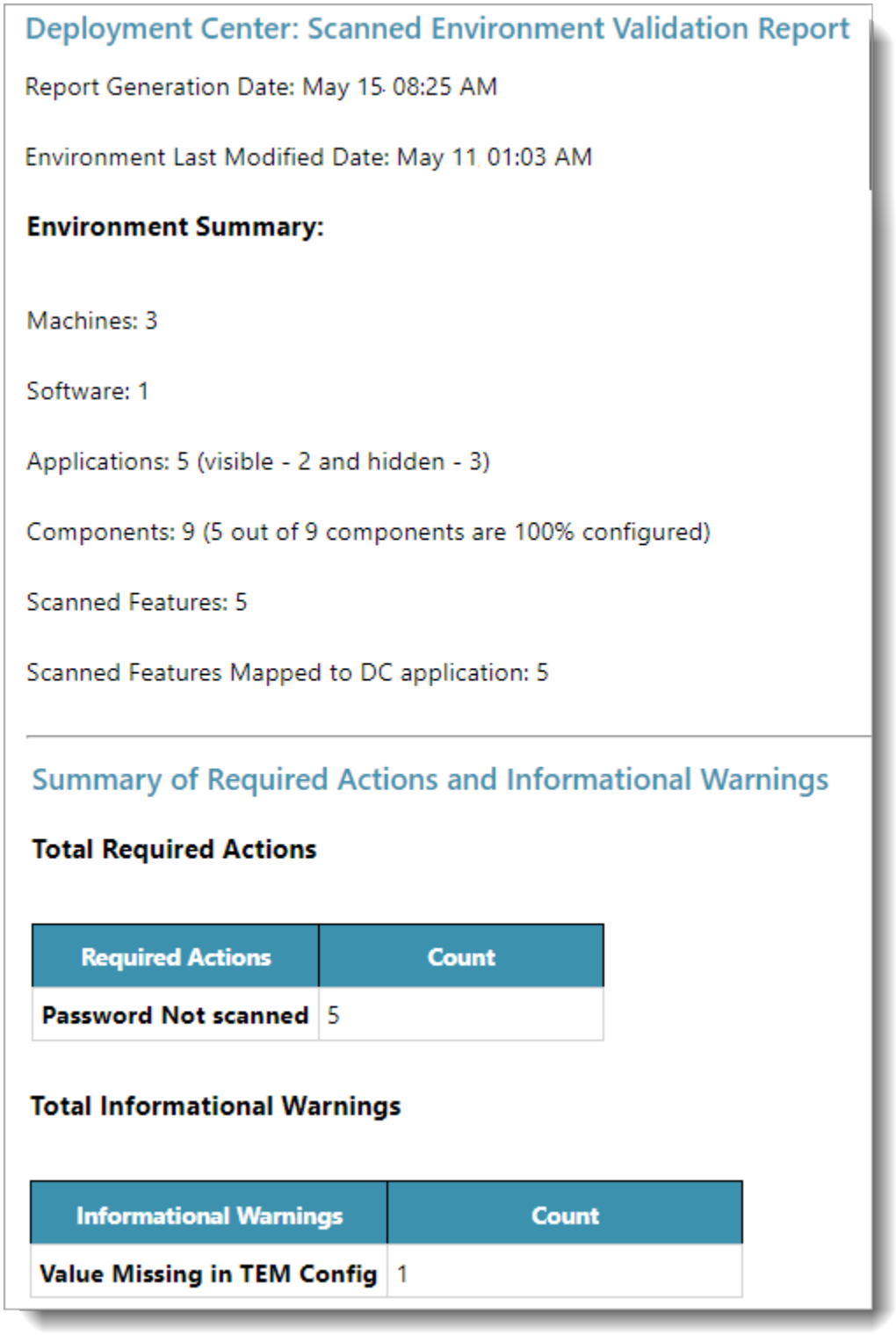
Deployment Center automatically downloads the report, so you can open it through the web browser **Downloads** feature or from the location above.

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.

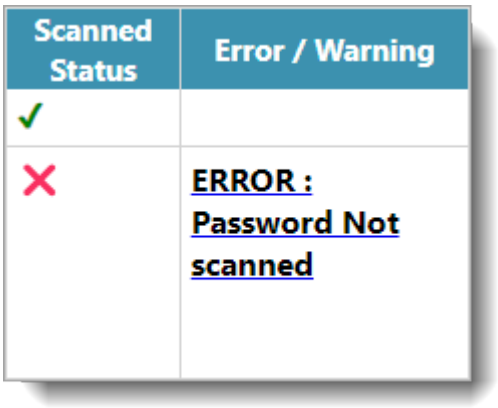
- **Mapping of Scanned TEM Environment Information for This Deployment Center Environment**

  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:

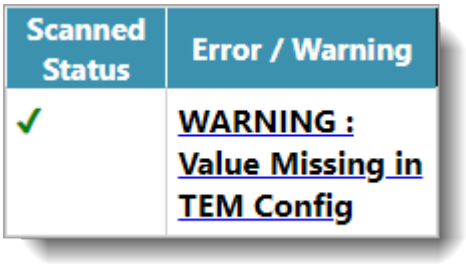
1. In each table in the report, note values in the **Scanned Status** and **Error/Warning** columns.

   In the **Scanned Status** column, a check mark (✔) indicates a valid item, an X () indicates an item that requires corrective action:

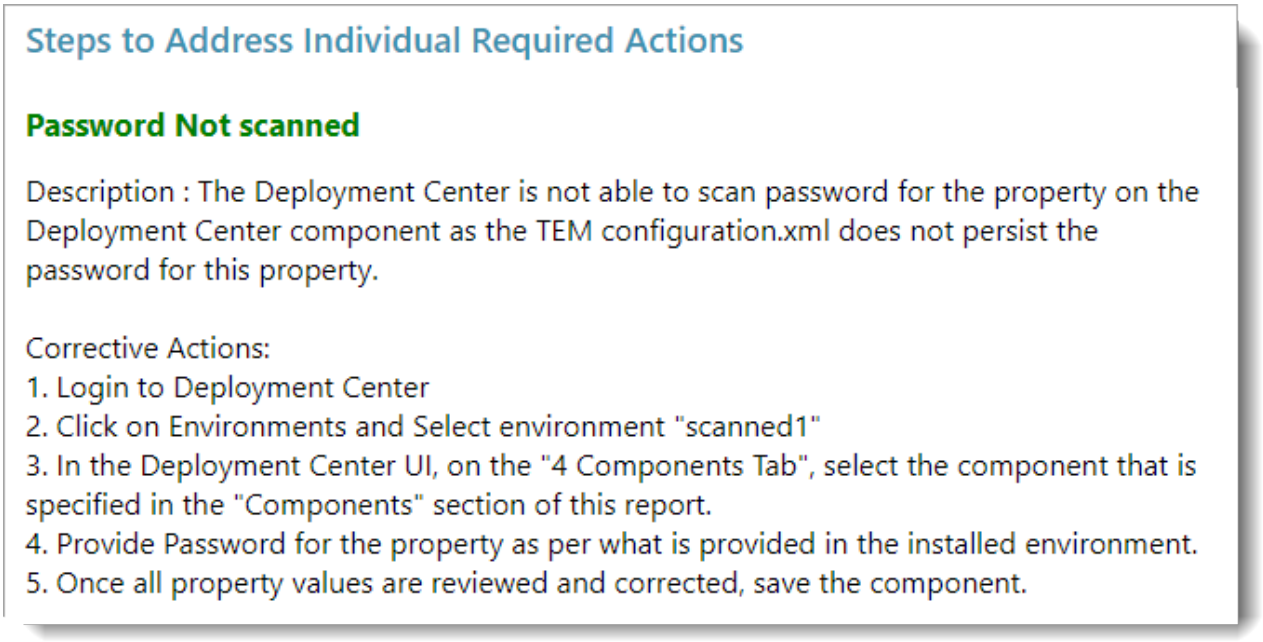


   Note:

   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 Troubleshoot registering environments.

Copy environments

You can create an copy of an existing environment from your **Environments** list in Deployment Center for various development or training purposes. Copying an environment duplicates the software,

applications, components, and configuration parameters of the selected source environment, changing the machine names to avoid impact to the source environment.

Copying an environment does not create a copy of the database. You must manually export the database and volumes.

> Caution:
>
> The **Copy Environment** function is not intended for copying an environment and deploy scripts to the *same* machines to redeploy the *same* environment. It should be used only to create a *new* environment from a source environment and deploy it on a *different* set of machines. Deploying a copied environment to the same machines as the source environment may result in ambiguous deployment status and the environment could become unrecoverable.

1. Log on to Deployment Center, and then click **ENVIRONMENTS**.

   The **Environments** page lists currently planned and registered environments.

2. In the environments list, select the environment you want to copy.

3. In the command bar on the far right, click **Copy Environment**.

4. In the **Copy source environment** dialog box, define values for the copied environment:

   a. In the **Environment Name** box, type a name for the copied environment.

   b. In the **Environment Type** box, choose the type of environment you want to create:

Development	A workspace where developers can perform experimental configuration and customization of software.
Integration	An environment in which multiple developers can integrate their code changes to validate the work of a team and ensure interoperability before committing changes to a test or production environment.
Production	A working environment with live users and data.
Test	A simulation of the production environment with the operating system infrastructure, software modules, custom code, and settings you intend to include in your production environment.
Training	An environment created for educational use that contains software modules, settings, and sample data for use in presenting or learning how to use the software. A training environment does not have to mimic a production environment, but can be a simplified environment that contains only selected components or applications.

Deployment Center performs no special configuration according to environment type. The **Environment Type** parameter is merely a tag to denote the purpose of the copied environment.

5. Click **Copy** to create a copy of your selected environment, or click **Cancel** to cancel the copy.

   Deployment Center adds the copied environment to the **Environments** list. You can continue configuring the copied environment at this time or return to it in a later session.

6. To continue configuring your copied environment, proceed to the **Applications** tab to add applications to, or remove applications from, your copied environment.

7. Proceed to the **Components** tab. Note any components whose **MACHINE** name is prefixed with **REPLACEME** and update those machine names as needed.

   Deployment Center inserts these machine names to prevent changes to the source environment.

8. Continue updating configuration parameters as needed on all components in the **Selected Components** list until the values in the **COMPLETE** column are all **100%**.

9. Proceed to the **Deploy** tab. Generate install scripts and deploy them on affected machines.

Remove environments

You can remove an environment from the **Environments** list in Deployment Center.

1. Log on to Deployment Center, and click **ENVIRONMENTS**.

   The **Environments** page lists all environments.

2. Select the environment you want to remove.

3. Click **Delete** on the command bar and confirm the removal.

The environment is only removed from Deployment Center tracking. The Teamcenter system remains intact.

> Note:
>
> Removing an environment removes Deployment Center tracking of the environment, including any stored snapshots of it. A removed environment can no longer be configured or deployed through Deployment Center. However, the Teamcenter system itself remains intact.

Back up and restore environments using snapshots

Deployment Center can create a *snapshot* of an environment, which records the applications, components, parameter settings, deployment status, and other properties in the form of a Quick Deploy XML file. Snapshots can be used to capture an environment in a current state and to roll back an environment to a previous state if necessary.

When you generate deploy scripts for an environment, Deployment Center automatically creates a snapshot of the environment before deployment scripts are run. Snapshots are stored in the Deployment Center repository.

> Note:
>
> A snapshot does not capture the entire Teamcenter deployment, such as the Teamcenter root directory and the Teamcenter database, it captures the *modeled* environment in Deployment Center. It is not used to restore a deployed Teamcenter installation, but the settings and contents of an environment before deployment.
>
> If you intend to use a snapshot to recover a deployment failure, back up your Teamcenter deployment, including database and volumes.

Using the **dc_quick_deploy** utility, you can **capture a snapshot** or **restore a snapshot** to roll back an environment to a previous state.

Before you create or restore a snapshot, you must **copy and unzip the dc_quick_deploy.zip file** from the Deployment Center installation.

Create a snapshot of an environment

To create a snapshot of a Teamcenter environment, run the **dc_quick_deploy** utility with the **create snapshot** argument:

dc_quick_deploy -create_snapshot -dcurl= *DC-URL* **-comment=***comment*
-environment=*environment-name* **-dcusername=***user* **-dcpassword=***password*

Example:

```
dc_quick_deploy -create_snapshot
  -dcurl=http://dc_host:8080/deploymentcenter
  -comment="Pending install Teamcenter 2412" -environment=env_tc2412
  -dcusername=dcadmin -dcpasswordfile="E:\admin passwords\dcadmin.pwf"
```

The utility creates the snapshot in the following location:

Windows systems:

*DC-repository***snapshot***environment-name_environment-ID***snapshot_***snapshot-ID***environment-ID*_**snapshot_***snapshot-ID*.**xml**

Linux systems:

DC-repository/**snapshot**/*environment-name_environment-ID*/**snapshot_***snapshot-ID*/*environment-ID*_**snapshot_***snapshot-ID*.**xml**

Note:

The **dc_quick_deploy** utility cannot create or restore a snapshot of a **global environment** (an environment with an **Infrastructure Type** of **Global**) or a local environment that references a global environment component.

Review environment snapshots

You can generate a listing of all existing snapshots for a certain environment by running the **dc_quick_deploy** utility with the **list_snapshot** argument:

dc_quick_deploy -list_snapshot -dcurl= *DC-URL* **-environment=***environment-name* **-dcusername=***user* **-dcpassword=***password* **-snapshot_file=***output-file-path*

In the **-snapshot_file** argument, specify an output text file for the listing.

Example:

```
dc_quick_deploy -list_snapshot
  -dcurl=http://dc_host:8080/deploymentcenter
  -environment=Env_tc2412 -dcusername=dcadmin
  -dcpasswordfile="E:\admin passwords\dcadmin.pwf"
  -snapshot_file=c:\\snapshots.txt
```

This command generates a report similar to the following:

```
Snapshot report for environment: Env_tc2412
Report generated on: Fri Jun 14 18:16:00 EST
Last snapshot restored*: Env_tc2412_snapshot_5
Snapshots available for Env_tc2412 stored in C:\PROGRA~1\DEPLOY~1\REPOSI~1\
   snapshot\Env_tc2412_81
|------------------|---------------------------------|--------------------|
| Snapshot         | Comment                         | Creation Date      |
|------------------|---------------------------------|--------------------|
| Env_01_snapshot_2 | Pending install Teamcenter 2412  | Apr 02 17:25:02 EST |
| Env_01_snapshot_3 | This is an automatic snapshot    | Apr 02 17:27:02 EST |
|                  | that was taken before deployment |                    |
| Env_01_snapshot_4 | Pending install Teamcenter 2412  | Apr 02 20:17:02 EST |
| Env_01_snapshot_5*| Pending install Teamcenter 2412  | Apr 12 12:44:02 EST |
| Env 01_snapshot_6 | This is an automatic snapshot    | Apr 12 17:59:02 EST |
```

```
|                   | that was taken before deployment |                     |
|-------------------|----------------------------------|---------------------|
```

In the **Snapshot** column, the asterisk (*****) indicates the last applied snapshot.

Restore a snapshot

To restore a snapshot of a Teamcenter environment, run the **dc_quick_deploy** utility with the **apply snapshot** argument:

dc_quick_deploy -apply_snapshot -snapshot=*snapshot-name* **-dcurl=***DC-URL*
-environment=*environment-name* **-dcusername=***user* **-dcpassword=***password*

Example:

```
dc_quick_deploy -apply_snapshot -snapshot=Env_tc2412_snapshot_2
  -dcurl=http://dc_host:8080/deploymentcenter
  -environment=Env_tc2412 -dcusername=dcadmin
  -dcpasswordfile="E:\admin passwords\dcadmin.pwf"
```

This command rolls back an environment to the state at which the snapshot was captured. Components that were partially configured in the snapshot are restored to their partially configured states.

Existing deployment scripts cannot be used after restoring a snapshot. You must generate new deploy scripts in the **Deploy** tab.

Note:

The **dc_quick_deploy** utility cannot create or restore a snapshot of a **global environment** (an environment with an **Infrastructure Type** of **Global**) or a local environment that references a global environment component.

Delete snapshots of an environment

You can use the **dc_quick_deploy** to delete old snapshots or snapshots you do not intend to use. Deleting a snapshot removes the snapshot file only. It does not remove backups of the Teamcenter root directory or database backups.

To delete a snapshot of a Teamcenter environment, run the **dc_quick_deploy** utility with the **delete snapshot** argument:

dc_quick_deploy -delete_snapshot -snapshot=*snapshot-name* **-dcurl=***DC-URL*
-environment=*environment-name* **-dcusername=***user* **-dcpassword=***password*

Example:

```
dc_quick_deploy -delete_snapshot -snapshot=Env_tc2412_snapshot_2
  -dcurl=http://dc_host:8080/deploymentcenter
  -environment=Env_tc2412 -dcusername=dcadmin
  -dcpasswordfile="E:\admin passwords\dcadmin.pwf"
```

Note:

When you delete an environment, Deployment Center automatically deletes snapshots of the environment.

Troubleshoot registering environments

The following sections provide information to help diagnose and resolve problems that occur when registering environments in Deployment Center.

Note:

For information about known software problems and workarounds, see the README files for Deployment Center and Teamcenter that accompany the software kits in the **Downloads** area on Support Center.

General environment registration failures

The environment registration utility (**send_configuration_to_dc**) generates log files when errors occur during environment scanning, such as:

- Sending configuration to Deployment Center fails.
- Inability to communicate with a Deployment Center server.
- Invalid credentials passed when sending environment configuration.

For details of errors during environment scanning, see these log files in the **send_configuration_to_dc_dir\logs** directory on the machine where you ran the utility.

- **tem_config_scanner_error_***timestamp***.log**

  Provides a description of environment scanning operation failures.

- **tem_config_rest_service_***timestamp***.log**

  Provides the communications information between the Deployment Center server and the **send_configuration_to_dc** utility.

Application dependency not scanned

This error occurs when a dependent application is not present in the **configuration.xml** from the TEM environment, so the application is not scanned.

The application that is not scanned is shown in bold text in the **Dependencies** column of the **Applications** table in the environment validation report.

To resolve this problem, you must manually add the missing application entry in the TEM **configuration.xml** file. This requires you to identify the global unique identifier (GUID) of the missing TEM feature by performing the following steps:

1. Copy the display name of the missing application dependency highlighted in bold text in the report. Search for this name in the following directory:

   *DC-repository***software***software-kit***install\lang\en_US**

   This search should return the **feature_***feature-name***_en_US.xml** file associated with the feature.

2. Open the **feature_***feature-name***_en_US.xml** file and locate the following line:

   ```
   <TextID name="feature-name.name" text="feature-display-name"/>
   ```

3. Search the *DC-repository***software***software-kit***install\modules** folder for the **feature_***feature-name***.xml** file.

   Note the *feature-name*.

4. Open the **feature_***feature-name***.xml** file and locate the line that contains the GUID:

   ```
   <guid value="GUID"/>
   ```

   Note this *GUID* value.

5. In the TEM **configuration.xml** file, locate the `<root><config><features>` section. In the area where other installed feature entries are listed, insert the following line:

   ```
   <installed feature="GUID" name="feature-name" />
   ```

Repeat these steps for any other "application dependency not scanned" errors.

> Note:
>
> Some dependencies may have been corrected in a later release. For example, the microservice package **oar1oaraw** is removed in later releases of Active Workspace, so this feature is reported as missing in scans with later releases.

If this happens, place the latest release of the software in Deployment Center repository, and then scan the environment again.

Component Missing

This error occurs when an expected component is not scanned. This typically occurs with the **Teamcenter Web Tier (Java EE)** or **Teamcenter Web Tier (.Net)** components. Deployment Center does not scan the web tier components, so you must add them manually to the environment after you scan the TEM environment into Deployment Center.

You can add the web tier components either of two ways:

- Prepare a **quick deploy file** to add the missing components, then apply the quick deploy to the scanned environment.
- Add the missing components in the Deployment Center user interface.

TEM Feature Not Scanned

This error occurs when a dependent application is not present in the **configuration.xml** from the TEM environment, so the application is not scanned.

This error occurs when a feature in the **configuration.xml** from the TEM environment is not scanned because the GUIDs of the TEM feature and the corresponding Deployment Center package file do not match.

Please report this error to product development to verify if this is already addressed in the later release of the product or not.

To resolve this problem, perform the following steps:

1. Note the TEM feature display name and the GUID for the feature not scanned. Copy the GUID from the **TEM GUID** column of the **Applications** table in the report.

   Search the *DC-repository***software***software-kit***install\modules** folder for the **feature_***feature-name***.xml** file.

   Note the *feature-name*.

2. Typically, the Deployment Center package file begins with the same feature name, for example, see the *feature-name***_package.xml** in the *DC-repository***software***software-kit***dc_contributions\packages** directory.

   If you cannot find the Deployment Center package file, search the steps executed in the TEM feature file, which should be the same as the steps executed in the Deployment Center package file:

a. In the TEM feature file, search for any copy or unzip operation, for example, the following line for the unzip operation in the feature file.

```
<unzip file="additional_applications/file.zip" todir=""/>
```

b. Search the *file*.**zip** in the *DC-repository***software***software-kit***dc_contributions****packages** directory to find the corresponding Deployment Center package file.

3. Correct the GUID in the Deployment Center package file to match the GUID in the TEM feature file.

```
<package guid="GUID-to-be-corrected" packageId="package-ID" >
```

Resolving connection errors

In the environment validation report, the **Required Actions for Network Connection Configurations** table shows all the unresolved connections between the components scanned in the environment. These problems are reported because the values for the properties scanned from the TEM **configuration.xml** mapped to the Deployment Center properties do not match. The table shows the TEM value in the **TEM Value** column and its Deployment Center value in the **DC Value** column. If the values in these columns do not match exactly, an unresolved connection error is reported.

To resolve these connection errors, you must make the values in the **TEM Value** and **DC Value** columns match by either of the following actions:

- Edit the TEM **configuration.xml** to modify the TEM value as to match the Deployment Center value, and then rescan the environment.
- Edit the Deployment Center property value in the Deployment Center user interface for the **Target Component** shown in the report.

  If a property such as **Machine Name** is disabled for modification, then you can supply the expected value of **Machine Name** in the **-machine** argument when you run the **send_configuration_to_dc** utility.

View registered machines

View the machines used in deployed Teamcenter environments. Select a machine name from the list to view its properties, such as operating system, disk capacity, and free disk space.

1. Log on to Deployment Center, and click **MACHINES**. The **Machines** page lists all servers used by components in deployed environments. When you select a machine, the **Overview** page provides information about the server where one or more software components are installed.

2. Verify that the properties for the server machine are what you expect.

Properties for a machine from a deployed environment include:

- **Machine Name** and **IP Address**

  Identifies the machine by the server name and IP address.

- **OS** and **OS version**

  Displays the operating system type and version installed on the machine.

- **Local Time**

  Displays the current date, time, and time zone at the machine's location.

- **Disk Capacity** and **Disk Free**

  Displays the total disk space and the free space available. The pie chart to the right displays the same information visually.

- **Last Update**

  Displays the last time information about this server was refreshed. The information is obtained and sent by the **send_configuration_to_dc utility**.

Maintain your environment

Deployment Center can maintain applications or components in a **registered Teamcenter environment**. Maintenance is making changes to your existing environment. You are not performing upgrades or applying patches. Before you perform maintenance, you must:

- Be sure you have a **registered existing environment** in Deployment Center before you perform maintenance on its components.
- **Send existing configuration information** about a registered environment to Deployment Center using the **send_configuration_to_dc** script.
- **Put your current environment's software in the repository** and check for dependencies.

Perform maintenance

The tabs for updating software or components are similar to installation:

1. Open the **Environments page** and choose the environment you want to maintain. Choose **Deploy Software** to begin the process.
2. **Software**

Choose software from the **Available Software** list. The software selections determine the list of available applications that you can update. Applications that must be updated are automatically selected. The **Selected Software** list displays currently installed versions.

3. **Options**

   Either single server or multiple server deployment is selected for **Environment Type**. If you previously had a **Single Box** environment, you can choose **Distributed**; however, you will need to update the server information for affected components.

   If an environment is already deployed on multiple servers, **Single Box** is not available.

   Architecture Type for your environment is automatically selected and can't be changed.

4. **Applications**

   Applications that are already installed are automatically displayed. You can add applications from the list. Applications that display a **Pending Install** status are waiting for deployment. Applications that are installed but need updates to support your selected software display the **Pending Update** status.

5. **Components**

   It's possible that a selection from the current update may cause a previously configured component to need more information, especially if you add applications or move from a **Single Box** environment to a **Distributed** environment.

   Components that are not yet installed display the **Pending Install** status. Components that are installed but need updates to support your selected applications display the **Pending Update** status.

   Components display the **%** configured. If it's less than **100%**, complete the required parameter values. Components that are not impacted can be ignored (showing **100%** configured).

6. **Deploy**

   Generate deployment scripts for the update. This tab is enabled when the other tabs are complete.

   Deployment scripts contain the information you configured in Deployment Center for the selected environment.

7. **Run the deployment scripts**

   After the scripts and software ZIP files are generated, copy them to each target machine and run them.

5. Deploying software using Deployment Center

Installation, upgrade, and maintenance strategy

You can use Deployment Center to install, upgrade, or update a Teamcenter environment. You can also perform maintenance on Deployment Center. You can choose the scenario that meets your needs.

Install a new Teamcenter environment

You can install software into a new environment. To prepare for a new installation:

- Define a Teamcenter environment in Deployment Center by one of the following methods:
  - **Create a new environment** if you are starting with a new Teamcenter installation.
  - **Register an existing environment** if you are installing new software in an existing environment. This may be the case if you are installing Active Workspace in an existing Teamcenter environment.
- **Put your unzipped installation software kits in a repository directory**. If you have an existing environment, put the software you used to install it into the repository.

Begin the installation by following the **deployment procedure**.

Upgrade or patch an existing Teamcenter environment

You can upgrade or patch software in an existing environment. Software dependencies are noted in the Deployment Center repository and in the **Software** deployment tab. Deployment Center displays messages explaining issues with upgrade software or dependencies.

To prepare for a software upgrade or patch in an existing environment:

- Be sure you have a **registered existing environment** in Deployment Center before you start an upgrade.
- Be sure to **send existing configuration information** from the environment to Deployment Center using the *send_configuration_to_dc* script.
- **Put your unzipped installation software kits in the repository** and check for messages about dependencies.

Maintain an existing registered Teamcenter environment

You can **update some software or components in an existing registered environment**. Maintaining an existing registered Teamcenter environment means updating software or components for your current version, as permitted by Deployment Center. Be sure you **send existing environment configuration information** to Deployment Center. You must have **the source software for your environment in the repository**.

Update Deployment Center configuration

You can run **deployment_center.bat -maintenance** to change the administrator password, add and remove repository directories, and add and remove deployment script directories.

Application deployment procedure

The Deployment Center **deployment process** walks you through a set of tabs that creates a Teamcenter environment and the deployment scripts containing software, application, and component information. This method shows you software and application choices, displays configuration information and parameters, and allows you to leave the user interface and return while you are entering your configuration information.

The Deployment Center user interface **Deploy Software** page displays each step in the deployment progress bar. In each tab, Deployment Center prompts you to make selections and provide information. At any time in the process, you can save your work and exit. The settings are stored in Deployment Center, and you can return to the deployment process at your convenience.

Dark blue means the tab is enabled and currently selected.

Medium blue means the tab is enabled.

Gray means the tab is not enabled yet. These tabs become available as steps within the previous tabs are completed.

Click a chevron to go to that tab. You can revisit any tab you previously completed to make changes. For example, if you completed the **Components** tab, you can still return to the **Applications** tab and make changes.

> Caution:
>
> Before proceeding to update an existing registered Teamcenter environment, be sure that you run the **send_configuration_to_dc script** from server components to provide current environment information in Deployment Center. Configuration changes performed locally on Teamcenter servers since the last time the **send_configuration_to_dc** script ran could be overwritten.

A Teamcenter environment doesn't track mass client software installations, so this action is not necessary for clients that have been deployed in this manner for the environment.

How to deploy software

1. Open the **Environments page** and choose the environment where you want to install or update software. The **Deploy Software** page provides access to the deployment tabs.

2. **Software**

   Choose the software to install or update. The software determines the list of available applications. For example, if you choose Active Workspace software, you can install the applications it provides.

   - **Pending Install** software will be installed or updated during deployment.
   - **Pending Update** software is already installed in your environment, but it needs an update to support other selected software.
   - **Installed** software is already installed in your environment and needs no updates.

3. **Options**

   Choose single server or multiple server deployment for **Environment Type**.

   Choose the Java or Windows architecture for **Architecture Type**.

4. **Applications**

   Choose the applications. The list of available applications depends on the software selected in the **Software** tab. Some applications may be automatically selected for you by default.

   Each software package can have one or more applications in its bundle. Applications provide business logic, data model, work processes, and administration data for specific business uses, industries, or integrations. In this step, you do not need to know details of your network or configuration of software or hosts.

5. **Components**

   Choose and configure components. Components run on the specified servers in your environment. Some components are automatically selected for you as required by the applications selected in the **Application** tab. You need to configure any component that is not listed as **100%** complete.

   Component status:

   - **Pending Install** components will be installed during deployment.

- **Pending Update** components are already installed in your environment, but they need updates to support your selected applications.

- **Installed** components are already installed in your environment, and they don't need any updates.

You need to know the server hosts on which components will be installed or updated, user names and passwords for the server component, and component URLs. Some components may have additional required or optional settings.

6. **Deploy**

   Generate deployment scripts. This tab is not available until the **Components** tab is complete.

   Deployment scripts contain the information you configured in Deployment Center for each of the servers in your environment. The scripts install the software, applications, and components onto each target machine in your environment.

7. **Run the deployment scripts**.

   After the scripts and software ZIP files are generated, copy them to each target machine and run them.

   If you are **installing both server and client deployments**, both types of scripts are generated. Always run server component deployment scripts before running client deployment scripts.

   You can run the deployment script in diagnostic mode to determine whether your script has any errors before updating the target machine.

Apply updates to your environment

Deployment Center software updates follow a process that is similar to installation. When you want to update your environment, choose the target version of software you want to apply, for example, Teamcenter 2412.0001. Deployment Center determines what to update based on what is required by the selected target release and selected applications.

Before you perform an update, you must download, unzip, and **put the software update or patch kits in the Deployment Center repository**. Check the repository for software dependencies and messages about missing software.

Import a configuration from TEM

Deployment Center can update or patch software in a **registered Teamcenter environment**. If the environment you want to update originated in TEM, run the **send_configuration_to_dc script** on the target servers to send the latest environment configuration information to Deployment Center.

You may not be required to put source release software kits in the repository. Deployment Center constructs source to target release mapping using the environment's current configuration files sent in the report from *send_configuration_to_dc*. Deployment Center analyzes the target release information to construct the mapping at the time you choose to update. Sometimes, Deployment Center may require the source software if the target release doesn't provide adequate mapping information. If there is missing software that is required, Deployment Center displays messages telling you about the dependency and how to proceed.

Perform updates or patches

The procedure for updating an environment is similar to creating an environment.

1. Open the **Environments page** and choose the environment where you want to update software. Choose **Deploy Software** to begin the process.

2. **Software**

   Choose the target update software from the **Available Software** list. The **Selected Software** list displays currently installed versions and latest pending versions for the environment. If the software you need is not available, check whether it was listed in the repository.

   If missing software is required, Deployment Center tells you about the dependency and how to proceed.

3. **Options**

   Either Single Box or Distributed is selected for **Environment Type**. If you previously had a **Single Box** environment, you can choose **Distributed**; however, you will need to update the server information for components.

   If an environment is already deployed on multiple servers, **Single Box** is not available.

   The **Architecture Type** for your environment is automatically selected and can't be changed.

4. **Applications**

   Applications that are already installed are automatically included for update. You can add other applications from the list. Applications that display a **Pending Install** status are waiting for deployment. Applications that are installed but need updates display the **Pending Update** status.

   See *Application names changed in Deployment Center* for additional information about application names.

5. **Components**

Components that are not yet installed display the **Pending Install** status. Components that are installed but need updates to support your selected applications display the **Pending Update** status.

It is possible that a selection from the current update may cause a previously configured component to need more information.

Components display the **%** configured. If it's less than **100%**, complete the required parameter values. Components that are not impacted can be ignored (showing **100%** configured).

6. **Deploy**

   Generate deployment scripts for the update. This tab is available when the **Components** tab is complete.

   Deployment scripts contain the information you configured in Deployment Center for the selected environment.

7. **Run the deployment scripts**

   After the scripts and software ZIP files are generated, copy them to each target machine and run them.

Server and client deployment scripts

In Deployment Center, you install or update server software by configuring their components. You can also configure client software components for mass client software installation. You generate both server deployment scripts and client deployment scripts using the **Deploy tab**.

Server deployment scripts

Select applications and configure the server components needed for your environment. After you **run deployment scripts on the target servers**, information about the target environment is sent to Deployment Center.

Server scripts must be run prior to running mass install client scripts for individual user machines. After the environment is updated, then run mass client install scripts on user machines.

If you are configuring client software as a component, such as a single installation of rich client on a specific server, this is considered a server deployment script because the deployment is for a specific server machine.

Mass client deployment scripts

Install and configure client software for users to run locally on individual machines. Client deployment scripts can only be created in a distributed environment. Mass client software is not machine-specific, so one client deployment script can run on any number of machines. Client scripts

do not send any information back to the Deployment Center, so the environment is not aware of which clients are installed or using it.

Multiple types of clients may be installed together with one script. Configure each type of client component and specify the same **Instance Name** for all of them. For every specified instance name, Deployment Center groups the installation scripts together and generates a single client install deployment script.

> Caution:
>
> Server component scripts must run prior to running any mass client installation scripts for an environment.

> Tip:
>
> If a user's machine already has a client installation and the deploy script can detect the installation path, the deploy script runs the update on the user's current client installation. If the deploy script has multiple clients but doesn't detect all the installation paths, it performs an installation if it doesn't find an installation path or an update if it finds the current installation.

Process for deploying mass client software

1. Client software deployment is performed as part of a **Distributed Teamcenter environment**. You may create mass client deployment scripts as part of configuring a Teamcenter environment.

2. Choose the client you want to deploy in the **Components tab**. If a client supports mass user installations, the configuration panel displays a check box to enable it.

3. Enable mass client deployment and specify the required parameters. **Machine Name** is replaced by **Instance Name** for a client script. Enter a name for the current client deployment.

4. Save the client component configuration and complete any remaining required component parameters.

   The client component displays **(Mass Client)** in the **Selected Components** list as the type of client deployment. Client components configured for mass deployment display a status of **n/a** because they are not tracked as part of a server component.

5. **Generate the deployment scripts.** The **Deploy Instructions** displays the path to the output zip files. A client deployment file is named *deploy_mass_client_instance-name.zip*.

6. Optionally, you can reduce the size of the client deployment scripts by running **the *generateMiniSoftware* utility**.

7. **Run the client deployment script after running any server component scripts** that were generated for the Teamcenter environment.

Client software may be installed as part of a server component where supported.

Generating mini-kits for mass client installations

You can reduce the size of your mass client deployment files by generating a *mini software kit*. Running the *generateMiniSoftware* utility extracts only the necessary software from the full software kits and builds a smaller, more portable software kit for the specific client deployment.

To generate a mini software kit, perform the following steps:

1. Locate the *minisoftwaregenerator.zip* file in the *additional_tools* directory in your Deployment Center installation. Expand this zip file to find the *generateMiniSoftware* utilities for Windows and Linux.

2. To generate a mini software kit, type the following command in a single line:

   ```
   generateMiniSoftware.bat -deployScriptDir=deploy-script-dir
   -softwareLocation=software-dir
   -outDir=output-dir -includeAllConfiguredApps
   ```

   Supply the following values in the command:

   deploy-script-dir: Path to the mass client deployment scripts directory. Be sure the scripts are in *.zip* file format.

   software-dir: Path to the directory or software repository from which the utility can gather files for the mini software kit. This can be a single path, or multiple paths separated by commas. Type the value the **default** to gather files from software kits in repositories registered with Deployment Center.

   output-dir: The location in which to generate the mini software kit.

   The optional **includeAllConfiguredApps** argument creates a mini kit that can be used to install a rich client *or* update an existing rich client. To create a mini kit for updating only, omit this argument. Omitting this argument creates a smaller but limited-use mini-kit.

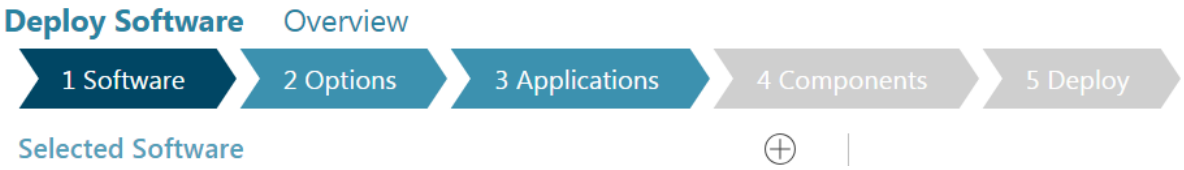
   For example:

   ```
   generateMiniSoftware.bat -deployScriptDir=D:\DC_deploy_scripts
       -softwareLocation=E:\DC_software -outDir=D:\mini_deploy_scripts
       -includeAllConfiguredApps
   ```

After you run the utility, its output log file contains information about which components and applications are included in the mini-kit as well as information about the size of the mini kit compared to the original software kits.

For more information about the *generateMiniSoftware* utility, see the *Teamcenter Utilities* in the Teamcenter documentation.

Software tab



In this tab, select the software to install from the list of installable applications. The software you select determines the list of applications available in the **Applications** tab. The **Selected Software** list displays both current and pending installations for the environment.

If a software kit you select depends on another kit, Deployment Center automatically selects the additional required kit.

1. In the **Software** tab, click **Edit Selected Software** ⊕ to add software.

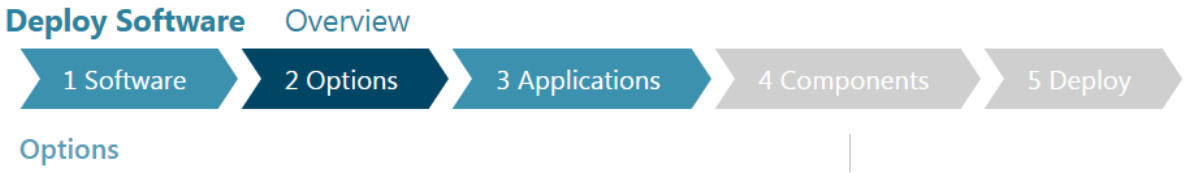
   This **Available Software** panel displays the software choices.

2. The **Available Software** panel lists software from the repository. The software status displays information about the software kit. After making your selections, click **Update Selected Software** to add them to **Selected Software**.

   If the software you need is not listed, you must **add it to the repository**. Add software as needed, but you may have to choose applications and configure components before deployment.

3. When your **Selected Software** list is complete, go to the **Options** tab.

Options tab



In this tab, choose the deployment options for your environment.

1. Choose the **Environment Type**.

   - Choose **Single box** to install all components on a single machine using the same installation path.

After you define **Machine Name**, **OS**, and **Teamcenter Installation Path** for one of the components, those values are adopted by the other components.

If an environment is already deployed on multiple servers, this type will not be available.

- Choose **Distributed** to install components on separate servers in an environment. This type may be selected automatically if your environment is already set up as a distributed environment.

  Key configuration values (**Machine Name**, **OS**, and **Teamcenter Installation Path**) are shared only with components that are required to be on the same server.

  If you install all components on the same host, but in *different installation paths*, choose the **Distributed** environment type.

You can change the value from **Distributed** to **Single box** if an install 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**.

   - Choose **Java EE** to filter component choices to the Java EE architecture.
   - Choose **.NET** to filter component choices to the Windows .NET architecture.

   If your environment already has deployed one of the architectures, the type is selected and can't be changed.

3. Choose **Infrastructure Type**.

   - Choose **Local** if your environment is to contain standard connected components or component configurations shared from a Global infrastructure environment. This is the default selection in a new environment.
   - Choose **Global** if your environment is to contain components configured for sharing to multiple Teamcenter Local environments.

     A Global infrastructure is used to define mass client information that can be shared to multiple environments managed in Deployment Center.

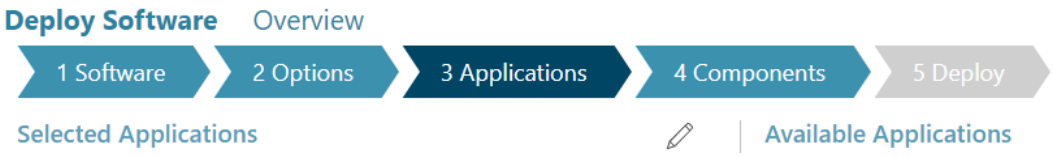
     Only certain components are supported for sharing to other environments, such as the four-tier rich client, TCCS, and Security Services Session Agent.

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 go to the **Applications** tab.

Applications tab



In this tab, choose applications for the software you selected. The list of available applications is determined by the **Selected Software** packages. Each software package includes one or more applications as a part of its bundle. The applications contain components, which you select later in the **Components** tab.

Some applications are automatically selected based on your **Selected Software**. For example, if you choose Active Workspace, the **Selected Applications** list displays the applications that are configured as required for installation.

Applications that have a **Pending Install** status are waiting for installation deployment to complete. Applications that have a **Pending Update** status are already installed but need an update to support other selections,

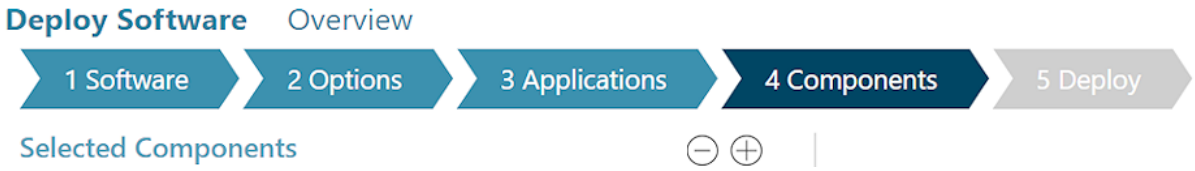
1. In the **Applications** tab, click **Edit Selected Applications** ⊕ to add applications.

   The **Available Applications** panel displays the application choices.

2. In **Available Applications**, choose the applications to install. If you choose an application that has one or more required applications associated with it, the associated applications are automatically selected. Click **Update Selected Applications** to add them to the **Selected Applications** list.

3. You can add or remove applications as long as they are not already installed. Selected applications show the **Pending Install** status. When your **Selected Applications** list is complete, go to the **Components** tab.

Components tab



In this tab, configure components for installation. Components provide the functionality for your environment. The **Selected Components** list displays required components that are automatically added from the **Selected Applications** list. **Selected Components** also displays optional components that were either already installed or previously selected. You can add more optional components from the **Available Components** panel.

Components that have a **Pending Install** status are waiting for installation deployment to complete. Components that have a **Pending Update** status are already installed but need an update to support other selections.

Some administrative tasks require that you have server names, user names, passwords, URLs, and other information available for the deployment. The following conditions may apply during component configuration:

- If a server machine was previously deployed in another environment or is specified in the current deployment for a different component, you can select it from the **Machine Name** list.
- If a component has a dependency on another component that is already defined, those values are shared with dependent components. This means that the component displays some percentage of completion.
- If you have not configured a component, the state may be either **Start** or some percentage complete if it has a shared dependency.
- When you are defining parameter values, some fields may not be editable. For example, if the component is already deployed in an environment, some parameters can't be changed (such as **Machine Name** and **OS**).
- If you selected the **Single Box** environment type in the **Options tab**:
  - Specifying a **Machine Name**, **OS**, and **Teamcenter Installation Path** for one component shares those values with the remaining components. If you change these values on one of the components, the changes are propagated to the other components when you save.
  - If a component is already selected or installed, it is only listed as an available component if multiple instances of that component are supported.

Add a component

1. In the **Components** tab, click **Add component to your environment** ⊕ to add components.

   The **Available Components** panel displays the optional component choices.

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

3. In **Selected Components**, the **COMPLETE** column displays the state of completion for required component settings.

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.

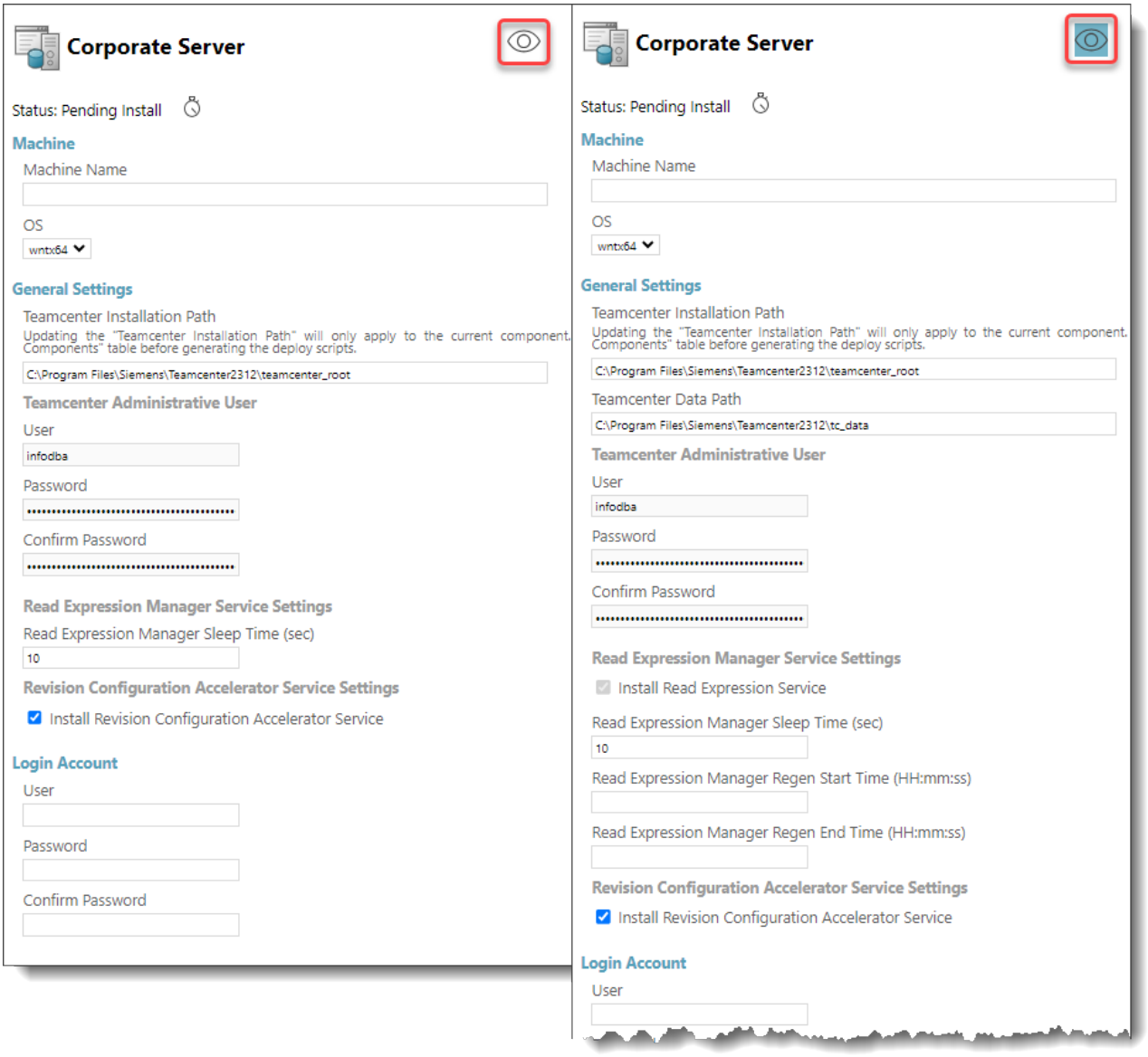
   Show all parameters

   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.



5. Completing all of the required settings pushes the state to **100%** complete. If you don't have all the information you need, you can save your settings at any time and return to finish them.

   For example, if you are installing the corporate server, required parameters include machine name, platform, Teamcenter installation path, and administrative user information. If you expand to **Show all parameters** , the corporate server displays additional optional settings.

6. When you are finished entering settings, click **Save Component Settings**.

7. The next component that is not complete appears in the right-side panel. When all **Selected Components** are **100%** configured, go to the **Deploy** tab. The **Deploy** tab is not enabled until the **Selected Components** are all complete.

Remove a component

You can remove components from an environment in the **Components** tab in Deployment Center, with certain conditions.

You can remove a component from the list, provided that the component:

- Is optional.
- Has a status of **Pending Install**.
- Does not have dependent components.

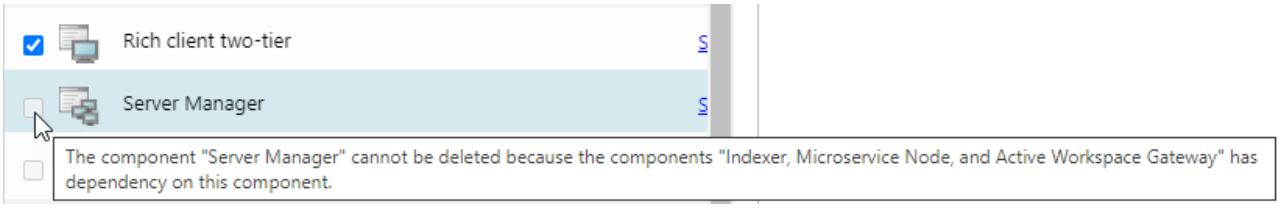
Deployment Center identifies components that can be removed.

1. Log on to Deployment Center.
2. In the **Environments** list, select the environment you want to modify.
3. Proceed to the **Components** tab.
4. In the **Selected Components** list, select the components you want to remove from your environment.

   The check box by each component name indicates whether the component can be removed:

Check Box	Description
☐	Enabled. Component may be removed.
☐	Disabled. Component may not be removed.
☑	Selected. Component is selected for removal.

   Hover over any disabled check box to view the reason the component cannot be removed. For example:



5. After you select components to remove, click **Remove Selected Components**, then confirm the deletion.

6. If any components do not have a configuration status of **100%**, update configuration parameters until all components show a configuration status of **100%**.

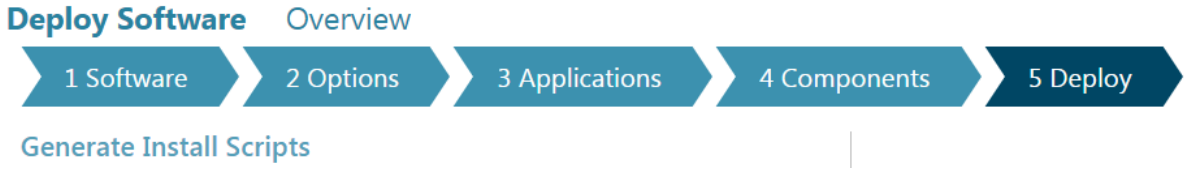
7. Proceed to the **Deploy** tab. Click **Generate Install Scripts** to generate deployment scripts to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

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*.

Deploy tab

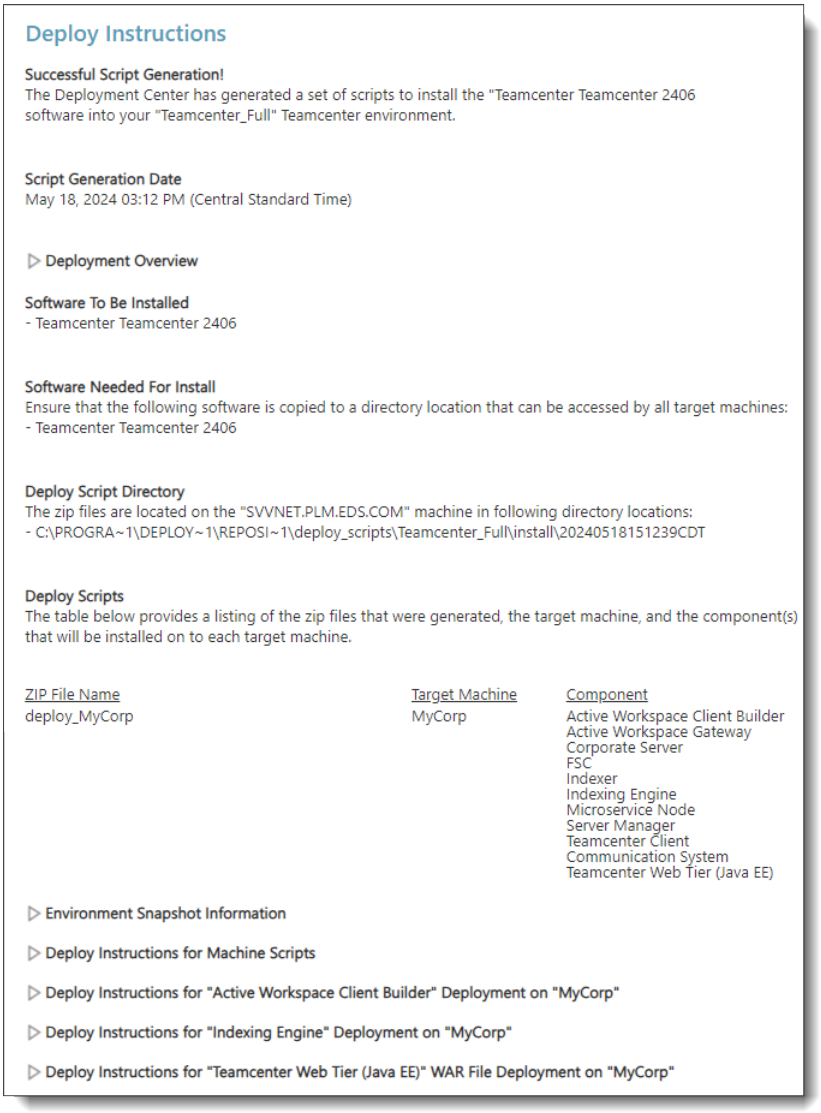


In this tab, generate deployment scripts for the components you want to install. Deployment scripts contain the information you configured in Deployment Center for the selected environment. These scripts install the software, applications, and components on to each target machine in your environment.

When the scripts are finished, the **Deploy Instructions** panel displays information about the deployment and instructions for proceeding with the deployment. You must copy the scripts to each target machine and run them to complete the installation.

1. To generate deployment scripts, click **Generate Install Scripts**.

   Deployment Center generates installation scripts, and reports information about the scripts in the right panel.



2. In the **Deploy Instructions** panel, you can view the report about the deployment, including the location of the deployment scripts and the instructions for continuing the deployment.

3. After you confirm that the scripts you need are in the directory, you can proceed with the deployment.

Description of the Deploy Instructions

- **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 *20180511_202452EDT__Sandbox_LM6W006.zip*, it runs on the **LM6W006** machine. Run an installation script only on its designated machine.

> Note:
>
> Deployment Center automatically generates an **environment snapshot** before generating deploy scripts. This snapshot can be used to restore the environment to its state before the deploy scripts were run.

Run the deployment scripts

As a best practice, run deployment scripts in diagnostic mode on the target machines to validate the configuration entered in Deployment Center. **Running each deployment script in diagnostic mode** tests the deployment configuration without making changes to the target machine during diagnostic validation.

The *deploy_scripts* staging area contains directories for each deployment environment. There is a subdirectory for each set of generated deployment scripts for the Teamcenter environment. Each server or mass client instance has its own deployment ZIP file.

The deployment files are generated to a location that was specified by the **-scriptsDir** configuration during **install** or **upgrade**, or **maintenance**.

The **dc_quick_deploy utility** may specify either a **-scriptsDir** value or a **-scriptDirectoryName** for the location of deployment scripts,

- *Deploy_Instructions.html*

  Provides information about the deployment and instructions for running your deployment scripts.

- Software ZIP files for the installation or update that contain the software required to complete your server or mass client deployment.

Caution:

When you deploy server components on an existing Teamcenter environment using the Deployment Center application, be sure that you run the **send_configuration_to_dc utility** to update the configuration information in Deployment Center before making changes. Configuration changes performed locally on Teamcenter servers since the last time the **send_configuration_to_dc** script ran could be overwritten.

Tip:

Be sure that your environment is not running when you deploy.

The **Deploy Instructions** in **the Deploy tab** display the path in the **Deploy Script Directory** section.

Be sure you logged in as a user with administrative privileges before running deployment scripts.

1. Navigate to the *deploy_scripts* directory. There may be one or more directories under the *deploy_scripts* directory that identify the environment with a timestamp subdirectory.

2. Determine which subdirectory you need, and find the ZIP files and the *Deploy_Instructions.html* that you generated.

   The server ZIP files have the naming convention:

   `deploy_`*`host_name`*`.zip`

   The client ZIP files have the naming convention:

   `deploy_mass_client_`*`instance_name`*`.zip`

3. Before you run the deployment scripts, make them available to the designated machines. For server deployments in your Teamcenter environment, copy the ZIP files to a directory that is accessible to the servers in your environment using one of these methods:

   Copy the ZIP files directly to each server and unzip them

   Select this method if you want to run the deployment locally on the machine. You must copy the correct ZIP file that matches each server. Be sure the server host name matches the *host_name* in the ZIP file name.

   Copy the ZIP files to a shared location, unzip them, and map a drive on each server

   Select this method if you want to run the deployment from a common location accessible to all the servers in the Teamcenter environment. You must share the deployment location by mapping a drive to that location on each server.

If you unzip on a Linux system, be aware that path and file names are in mixed case. Avoid converting path and file names to lowercase, as paths are case sensitive. See the documentation for your ZIP utility for information.

> Caution:
>
> Be sure to run server deployment scripts before running client deployment scripts for an environment.

4. Set these environment variables to the location of a certified 64-bit Java JDK:

   JRE64_HOME
   JRE_HOME
   JAVA_HOME

5. If a Teamcenter server manager is running, stop it.

6. Open a command prompt window and navigate to the location where you unzipped the files.

7. Run the *deploy.bat* (Windows) or *deploy.sh* (Linux) script with the necessary arguments:

Argument	Description
-dcusername	Specifies the user name of the Deployment Center administrator. You defined this user when you installed Deployment Center.
-dcpassword OR **-dcpasswordfile**	Specifies the password for the Deployment Center administrator. Use **-dcpassword** to specify the password as text, or use **-dcpasswordfile** to specify **an encrypted password or password file**. If the password file path contains spaces, enclose it in quotes.
-softwareLocation	Specifies the software location or locations. Set to **default** to find software in the registered repository directories, or specify one or more software paths separated by commas. For example: `-softwareLocation=D:\deploy_software,X:\Foundation_software` The software directories specified must include *software* subdirectories.
-dcurl (optional)	Specifies the URL to the Deployment Center server, if it differs from the URL specified in the **deploy** script. When you run a deployment script on a server that is on a different network than the Deployment Center server, you can specify an alternate URL for Deployment Center. You may specify a fully qualified domain name, an alias, or a reverse proxy for the server in the URL.

Argument	Description
	The specified URL replaces the Deployment Center URL in the **deploy** script configuration for authentication and communication with the Deployment Center server.
-diagnosticChecks (optional)	Specifies diagnostic mode. This option runs a diagnostic validation test of the deployment. Diagnostic mode checks whether the deployment tasks in the **deploy** script can be completed successfully on the target machine. Diagnostic mode does not perform any updates during validation. Run the deploy script in diagnostic mode to validate properties like operating system and database credentials, ports, installation paths, FSC unique IDs, and so on. **The log output provides success and failure information**. For client mass installation scripts, there is no server host validation as they are machine neutral. Make any necessary corrections in Deployment Center, regenerate your deployment scripts, and run them again in diagnostic mode. Repeat until all the errors are addressed.
-ignoreDiagnostics (optional)	Skips diagnostic validation tests of the deployment. You may want to ignore diagnostic validation if you determine that an issue reported by **-diagnosticChecks** does not need to be addressed. Specify this argument if you've received a diagnostic report that meets your requirements for acceptable deployment.
-startServices	Starts the services of all required components in an installed environment. Run the deploy script in this mode to start all the necessary services. Running the deployment with this option *only* performs the service start operation and skips executing the remainder of the deployment tasks.
-stopServices	Stops the services of all required components in an installed environment. Run the deploy script in this mode to stop all the necessary services. Running the deployment with this option *only* performs the service stop operation and skips executing the remainder of the deployment tasks.
-getServiceStatus	Retrieves the status of the services of all required components in an installed environment. Run the deploy script in this mode to retrieve the status of all the necessary services. Running the deployment with this option *only* performs the service status operation and skips executing the remainder of the deployment tasks.

On Linux systems, make sure you run the script in Korn Shell (**ksh**) to avoid errors. For example:

```
./deploy.sh -softwareLocation=/kits/software
-dcusername=dcadmin -dcpasswordfile=/opt/passwords/dcadmin.pwf
-dcurl=http://dcserver.mysite.com:8008/deploymentcenter
```

When the installation or update is complete, the command prompt returns a success message with the location of the log files.

If you experience a problem in running the **deploy** script, see **Troubleshooting deployment**.

If you have multiple Teamcenter servers in your environment, you can run deployment scripts on them in parallel. To do this, run the deployment script on the primary business logic server (the corporate server) first. Deployment Center uploads a dataset to the volume that contains key items from the **TC_DATA** directory, which provides the necessary information for deploying on the remaining servers. You can then subsequently run the install or update deployment script on the remaining servers concurrently. Running deployment scripts in parallel can significantly reduce deployment time.

How to deploy using a software mini-kit

If you **generated a software mini-kit**, you can run the mass client deploy script using the mini-kit by performing the following steps:

1. Copy the mini-kit zip file to a directory accessible to the target machine on which you will run the mass client deploy script.

   The mini-kit file is named *env-name_mass-client-instance_OS*.**zip**, for example, **Env001_TcMassClient_wntx64.zip**.

2. Expand (unzip) the mini-kit in the location to which you copied it.

3. Copy the mass client deploy script (*deploy.bat* on Windows or *deploy.sh* on Linux) to the target machine.

4. Run the mass client deploy script as for a typical installation, except in the **softwareLocation** argument, specify the location of the mini-kit you expanded in step **2**.

Running a mass deploy script using an automation tool

If you run a mass deploy script using an automation tool like Microsoft System Center Configuration Manager (SCCM) that executes scripts from a system account when no user is logged in on the given machine, disable User Account Control (UAC) to enable the script to run:

1. Open the **deploy.bat** file for the given machine in a text editor.

2. Remove or comment out the following lines:

```
call "%DEPLOYER_DIR%\dc_util\check_uac_status.bat"
if "%errorlevel%"=="1"(
    exit /b 1
)
```

3. Save the changes to the file.

This enables the deploy script to skip UAC validation during deployment.

Troubleshooting deployment

Troubleshooting deployment on a target machine

Three types of log files may be generated when you run the deploy script on a target machine. If an error occurs when you run **deploy.bat** or **deploy.sh**, first review the log file **deployer_***timestamp***.log**.

- **Deployment Center logs**

  Deployment Center log files are located in the **logs** directory under your unzipped **deployment_script_dir** directory on the Teamcenter machine where you are running **deploy.bat** or **deploy.sh**.

  Review the **deployer_***timestamp***.log** log file created by the **deploy.bat** or **deploy.sh** scripts and check for errors.

  Some of the more common errors you may see:

  - Failure to locate the software required by the script as specified by **-softwareLocation**.
  - Failure to communicate with a Deployment Center server.
  - Other deployment failures. These may be related to the machine, the environment, Deployment Center, the software's contributions for Deployment Center, Teamcenter utilities, or Business Modeler IDE.

  Look for a section titled **Diagnostic Checks Details**, which provides validation information.

  If you have **Diagnostic Checks** errors, make the corrections using the method you used to create the deployment script. Regenerate the deployment scripts and run them again in diagnostic mode. Repeat this process until all errors are addressed.

  After **Diagnostic Checks** have passed, deployment errors shown in the **deployer*.log** files may also require reviewing logs from the Teamcenter utilities that are called by Deployment Center.

- **Teamcenter utility logs**

  Deployment Center deploy scripts may call Teamcenter utilities to perform actions in the Teamcenter database. These utilities (for example, Teamcenter installation utilities) try to connect to servers during deployment.

  Logs for these utilities are located in either the **logs** directory under *TC_ROOT* or the server's temporary directory (for example, **TEMP** or **TMP**).

- **Business Modeler IDE logs**

  The Business Modeler IDE utilities are run by Deployment Center and can encounter an error during deployment. This information is shown in the **deployer*.log** files, but additional information may be recorded in Business Modeler IDE logs located in the **logs** directory under the *TC_ROOT* directory.

Deployment Center assists in troubleshooting by automatically retrying failed commands that were called by a deploy script. When a command called by Deployment Center fails during a deploy action, the deploy script automatically enables the debug mode for the Teamcenter utility that was called, and then retries the command that failed.

Running a Teamcenter utility in debug mode provides additional troubleshooting information in the utility's debug log. If the given command succeeds when it is retried, Deployment Center turns *off* debug mode so the utility does not generate a debug log file. This minimizes clutter in the **logs** directory, avoids creation of large unnecessary log files, and thereby helps focus troubleshooting efforts.

For information about utilities and debug modes, see Teamcenter Utilities.

6. Quick deployment

Understanding quick deployment

An alternative to using the Deployment Center application is to deploy software using a utility. The utility uses an XML configuration file to provide the software, application, and component information for your target environment. You can install a new Teamcenter environment or update an existing one.

Quick deployment considerations

Consider using quick deployment if:

- You want to deploy a simple environment quickly.
- You want to make regular scripted updates to an environment.
- You want to generate test or development environments frequently.
- You want to generate new environments from existing environments.
- You want to schedule or automate environment updates using continuous development tools.

The quick deployment approach

The principles for setting up an environment and choosing software in the Deployment Center application also apply to the quick deployment process. You must have experience setting up and configuring applications and components for Teamcenter environments to effectively use this approach.

The **quick deployment procedure** explains how to:

- Create the XML configuration file that contains application and component configuration for your environment. You can export an **XML configuration file** from an existing environment to use as a starting point. Update the file with the configuration required to install or update your Teamcenter environment. XML configuration files can be generated from the Deployment Center application **Environments** page or the **dc_quick_deploy** utility.

  You may want to use an existing environment that was created using Teamcenter Environment Manager (TEM) as your starting point. **Register the environment** using the **send_configuration_to_dc** utility to send the configuration information to Deployment Center. Then you can generate an XML configuration file from the environment for editing.

- Create deployment scripts for the target environment by running the **dc_quick_deploy utility**. The input is your XML file containing your software, application, and component configuration. The utility generates the deployment scripts in the directory specified by the **-scriptsDir** parameter. Configure the locations for the **-scriptsDir** parameter during **install**, **upgrade**, or **maintenance**.

- **Run the deployment scripts** generated by **dc_quick_deploy** on the target component servers, as you normally would, to complete the installation or update.

Quick deployment procedure

The **quick deployment process** creates or updates a Teamcenter environment using deployment scripts generated by the **dc_quick_deploy utility**. The process uses an XML configuration file to provide application and component information rather than obtaining it directly from the Deployment Center application.

This type of deployment requires that software, application, and component configuration information are defined in **a structured XML configuration file**. The **dc_quick_deploy** utility processes the XML file, and then generates deployment scripts and instructions.

If you are also deploying mass client installations or updates, the **dc_quick_deploy** utility generates **two types of deploy scripts**, one for server software and one for client software.

Define your deployment

Before you begin, define the components for your software and applications. Gather and evaluate the associated server information as needed for each component. You must have experience setting up and configuring applications and components for Teamcenter environments to successfully use the quick deploy approach.

Ensure the repository has the software required for your installation or update. Identify and include any software dependencies. Determine whether you are using a single or distributed server environment. Define the list of applications and their dependencies.

You may need to know server names, locations, operating systems, port numbers, user names, passwords, installation paths, connection URLs, or other required or optional settings for your components.

You can obtain software configuration information in a couple of ways:

- You can review software configuration information from the **Repositories** page for each software kit. From the **Active Media** tab, click **Generate Software Configuration Report**  . The HTML report provides comprehensive information about software dependencies and IDs, application internal names and dependencies, and component and property internal names and configuration values.

- You can create a quick deploy **XML configuration file** from an existing Teamcenter environment and edit it for reuse in another environment.

  Information from the **Software Configuration Report** may be helpful in making updates to the XML configuration file.

1. Assemble the information required to successfully deploy the target Teamcenter environment, including software, application, server, and configuration settings.

2. Create an XML configuration file to work from. You can **evaluate the best method to create or obtain an XML configuration file**.

   You may prefer to start from an existing Teamcenter environment configuration:

   - If the environment is registered in Deployment Center, select it from the **Environments** page and choose **Export Environment**.

   - Run the **dc_quick_deploy** utility in **export mode** to generate an XML configuration file from a specified environment.

   - For an environment that was configured using TEM, **register the environment** with Deployment Center using the **send_configuration_to_dc** utility. Then you can export an XML configuration file.

3. Edit the XML configuration file, ensuring it contains all the configuration information for your desired environment.

   You will need to provide passwords for target servers in the configuration file. The **dc_quick_deploy** utility can generate **encrypted passwords** for you if needed. The encrypted password can then be entered in the XML configuration for the specific component.

4. After you complete the XML configuration file, run the **dc_quick_deploy** utility in **import mode** to generate deployment scripts. The import mode requires the XML configuration file you created as input.

   The output includes generated deployment scripts for each of the servers in your environment and a **Deploy Instructions HTML file**.

   If **dc_quick_deploy** encounters an error, the returned error message includes information about how to resolve the problem.

   If you are generating both client and server scripts, the output provides:

   - One or more server scripts, where the zip file name follows the pattern **deploy_***server-name***.zip**.

   - One script for each set of client installs, where the zip file name follows the pattern **deploy_***mass_client***.zip**.

5. After the script files and instructions are generated, copy them to each target machine and run them as you usually would. Run **the deployment script in diagnostic mode** on the target machine to determine whether you have errors. After making your corrections, run the deployment script.

Using the dc_quick_deploy utility

The **dc_quick_deploy** utility provides the same functionality as the Deployment Center application for creating or updating a Teamcenter environment. The Quick Deploy command line method supplies environment information to Deployment Center in an **XML configuration file**. The **dc_quick_deploy** utility processes the configuration information to generate deployment scripts for the target machines. The **quick deployment procedure** guides you through producing the XML configuration file and generating the deployment scripts.

Find the Quick Deploy utility package in the following location in your Deployment Center installation:

Windows systems:

Deployment Center-installation\webserver\additional_tools\dc_quick_deploy.zip

Linux systems:

Deployment Center-installation/webserver/additional_tools/dc_quick_deploy.zip

Copy the package to a local machine, unzip it, and locate the Quick Deploy utility (**dc_quick_deploy.bat** on Windows systems, **dc_quick_deploy.sh** on Linux systems).

The utility performs several deployment actions, such as the following:

- **Export a configuration as a baseline for a new environment**
- **Import a configuration to a new environment**
- **Encrypt passwords for XML files**
- **Create a snapshot of an environment**
- **Apply a snapshot to roll back an environment to a previous state**

To view usage details for the utility, type **dc_quick_deploy -help**.

Export mode

The **export** mode exports the configuration information for an existing environment in Deployment Center. Use this mode to generate a valid XML configuration file that you can use as a starting point for a new or updated environment. The following arguments are *required* when you run the **dc_quick_deploy** utility in export mode:

Argument	Description
-mode (required)	Specifies **export** to generate the environment's configuration.
-environment (required)	Specifies the name of the Teamcenter environment being exported.
-exportfile (required)	Specifies the full path and name of the file, including the *.xml* extension, where the configuration file will be exported.
-exportType (required)	Specifies whether the exported file contains **Full (all component properties) or Lean (only required properties)** configuration information.
-dcURL (required)	Specifies the URL to access Deployment Center.
-dcusername (required)	Specifies the user name for Deployment Center as defined when installing Deployment Center.
-dcpassword *or* **-dcpasswordfile** (one is required)	Specifies the password text *or* the **encrypted password or file**.
-preserveDeploymentStatus (optional)	Specifies you want to export the environment preserving the deployment status of software, applications, and components.

Example:

```
dc_quick_deploy.bat -dcurl=http://abc34xyz:8080/deploymentcenter
   -mode=export -environment=test_env -exportType=Full
   -exportfile=D:\export_files\quick_deploy.xml
   -dcusername=dcadmin -dcpassword=14pXjKcFG8CIrbjdy1teV5JWJY
```

Import mode

The **import** mode imports a configuration file containing the configuration for the specified environment. Use this mode to import a valid XML configuration file for a new or updated environment. The following arguments are available when you when you run the **dc_quick_deploy** utility in import mode:

Argument	Description
-mode (optional)	Specified by default, **import** imports the specified XML configuration file.
-environment (required)	Specifies the name of the Teamcenter environment being created or updated.
-inputFile (required)	Specifies the path and file name for a valid environment XML configuration file.
-dcURL (required)	Specifies the URL to access Deployment Center.

Argument	Description
-dcusername (required)	Specifies the user name for Deployment Center as defined when installing Deployment Center.
-dcpassword *or* **-dcpasswordfile**	Specifies the password text *or* the **encrypted password or file**.
-machine (optional)	Specifies the name of the machine where the deployment scripts will be generated, which defaults to the current user's machine if not specified.
-platform (optional)	Specifies the operating system for the machine where the deployment scripts will be generated, which defaults to the current user's platform if not specified. For example, use **wntx64** for Windows and **lnx64** for Linux.
-scriptDirectoryName *or* **-scriptsDir** (optional)	Specifies the path to where deployment scripts will be generated. **-scriptDirectoryName** specifies a directory name using letters, numbers, and underscores. No special characters or spaces are supported. **-scriptsDir** specifies the parameter that represents the path where you want to generate deployment scripts. The parameter was defined during Deployment Center **installation**, **upgrade**, or **maintenance**. For example, **-scriptsDir=scriptsDir123** resolves to the file path that was configured for **scriptsDir123**. If **-scriptsDir** resolves to a UNC path, Deployment Center must be running as a Windows service.
-override (optional)	Specifies the path to an **XML override** file or comma-separated list of paths.
-create_snapshot	Specifies you want to **create a snapshot** of the given environment.
-delete_snapshot	Specifies you want to **delete a snapshot** of the given environment.
-list_snapshot	Specifies you want to **list all snapshots** of the given environment.
-apply_snapshot	Specifies you want to **apply a snapshot** (restore to a previous state) of the given environment.

Example:

```
dc_quick_deploy.bat -dcurl=http://myCorp:8080/deploymentcenter
   -environment=test_env -platform=wntx64 -machine=abc236wxyz
   -inputFile=D:\import_files\quick_deploy_input.xml
   -dcusername=dcadmin -dcpasswordfile=C:\dc_pwds\deploymentcenter.pwf
```

Encrypt passwords for your server components

You can use **dc_quick_deploy** to quickly generate encrypted passwords that you can use to specify passwords in the XML configuration file.

```
dc_quick_deploy -encrypt=clear_text_password
```

Enter the resulting encrypted password in the property specification for a component, as in this example:

```
<property id="fnd0_tcAdminPassword" value="'lE6tFf61dcrRZHuMzJ34Sx2a"
   encrypted="true"/>
```

Creating an XML configuration file

The **dc_quick_deploy** utility requires an XML file containing the configuration information to create deployment scripts for the desired environment. The XML configuration file contains an environment's settings for software, applications, and components.

The easiest way to create an XML configuration file is to export the current configuration of an existing environment that most closely resembles the one you wish to produce. You can then edit the exported XML file to add or update the configuration information for your desired environment. After you complete your edits, run **dc_quick_deploy** again, and import the XML configuration file as the input file. This action generates the scripts you deploy on the target servers.

You can find **examples of XML configurations files** to assist you.

Choose an export method

There are a couple of ways to export an environment's XML configuration file:

- On the **Environments** page, choose an environment from the list, and click **Export Environment**.

  Select whether you want the output XML file to contain all component configuration properties or just the required component properties.

- Use **the export mode** of the **dc_quick_deploy** utility to generate an XML configuration file for a specified environment.

  Choose whether you want to generate a **full** (all configuration properties) or **lean** (only required configuration properties) XML file.

Choose either Full or Lean configuration

As part of a strategy for creating Teamcenter environments, evaluate what the XML configuration file requires. When specifying the **exportType**, you can choose **Full** or **Lean**.

Full

Includes all the application and component parameters for the environment.

Lean

For applications:

- Includes only top level applications that display in the Deployment Center application.

  Does not include applications associated by default with a software selection, such as Teamcenter Foundation and Active Workspace.

  Does not include applications that are either automatically selected as dependent applications or hidden in Deployment Center.

For components:

Includes only required properties for components, including properties that have a value different from the default value.

Import the XML configuration file

After you update the XML configuration for your desired environment, use the **the import mode** of the **dc_quick_deploy** utility to generate the scripts to create the environment from the input XML configuration file.

Find examples of quick deploy XML configuration files

The Teamcenter Deployment Reference Architecture contains sample Quick Deploy and configuration files that you can review to learn more about the XML file structure.
Open the downloads page for Teamcenter 2412 and download the latest version of the **Teamcenter_Deployment_Reference_Architecture_*version*.zip** file.

Also, see the **Software Configuration Report** for information that may be helpful in making updates to the XML configuration file.

Using an XML override file

The **dc_quick_deploy** utility supports overriding certain properties in a Quick Deploy file using an override XML file.

An override file specifies property values to override from the XML configuration file, such as ports, user names, and machine names for certain components. This is useful when you deploy the same configuration on multiple machines with only slight variations.

For example, consider an XML configuration file that contains the following properties:

```
<component id="aws2_indexingengine" machineName="server1"
platform="wntx64">
            <property id="aws2_indexingEngineUserPassword"
value="Password1"  encrypted="false"/>
            <property id="aws2_machinePassword" value="Password1"
encrypted="false"/>
            <property id="aws2_machineUser" value="server1\user1" />
</component>
```

To override the machine name and password from the XML configuration file, create an XML override file with the following properties:

```
<component id="aws2_indexingengine" machineName="server2"
platform="wntx64">
            <property id="aws2_indexingEngineUserPassword"
value="Password2"  encrypted="false"/>
</component>
```

Save this file as **quick_deploy_override1.xml**. Then, specify this override XML file when you run the **dc_quick_deploy** utility. For example:

```
dc_quick_deploy.bat -dcurl=http://myCorp:8080/deploymentcenter
   -environment=test_env -platform=wntx64 -machine=server1
   -inputFile=D:\import_files\quick_deploy_input.xml
   -dcusername=dcadmin -dcpasswordfile=C:\dc_pwds\deploymentcenter.pwf
-override=C:\quick_deploy_override1.xml
```

The **override** argument can specify multiple override XML files, separated by a comma. For example:

```
dc_quick_deploy.bat ...
-override=C:\quick_deploy_override1.xml,C:\quick_deploy_override2.xml,C:\
quick_deploy_override3.xml
```

The paths specified in the **override** argument must be absolute paths.

An override XML can change properties of components included in a Quick Deploy input file, but it *cannot* change the general composition of the environment, such as the following attributes:

- Applications (cannot add or remove applications)
- Components (cannot add or remove components)
- Software (cannot add or remove software)
- Architecture type (Java EE or .NET)

Sharing components across environments using Quick Deploy

Using Deployment Center, you can create an environment that contains client components that can be shared across multiple Teamcenter environments. By setting the **Infrastructure Type** option in Deployment Center to **Global**, you create an environment whose components can be imported into environments whose **Infrastructure Type** is set to **Local**.

The following client components are supported for sharing from a Global infrastructure environment:

- **Rich client four-tier**
- **Teamcenter client communication system**
- **Teamcenter Security Service Session Agent**

You can create a Global or Local environment using Quick Deploy.

The following example defines a Global infrastructure environment. The **infraTypeName** tag designates the environment as **Global**. The **infrastructureType** properties indicate a component belongs to a Global environment and is therefore sharable.

```
<?xml version="l.0" encoding="UTF-8" standalone="yes"?>
<quickDeployConfig configName="Env_006" version="l.0">
    <archType types="J2EE"/>
    <infraType infraTypeName="GlobalEnvironment"/>
    <quickDeploySoftware>
        <software id="Foundation" version="2312.20231030.00"/>
    </quickDeploySoftware>
    <quickDeployApplications/>
    <quickDeployClients>
        <client id="fnd0_4tierrichclient" machineName="testingl"
massDeploy="false" platform="wntx64
            <property id="fndO_compressWebResponse" value="true"/>
            <property id="fnd0_4tierrichclient.massDeployCheck"
value="false"/>
            <property id="fnd0_4tierrichclientInstallationPath"
value="C:\Program Files\Siemens\Teamcenter\
                teamcenter root"/>
            <property id="fnd0_4tierrichclient.appSupportedList"
value="selectedApps"/>
            <property id="fnd0_4tierrichclient.infrastructureType"
value="GlobalEnvironment"/>
            <property id="fndO_racRuntimeTempFolder" value=""/>
            <property id="fnd0_4tierrichclientPortalViewerLicenseLevel"
value="Base"/>
            <package id="fnd0_foundation"/>|
        </client>
    </quickDeployClients>
    <quickDeployComponents/>
    <removeQuickDeployApplications>
```

```
    <package id="fnd0_foundation"/>
    </removeQuickDeployApplications>
```

After configuring the Quick Deploy script for your Global infrastructure environment, you can deploy it using the **dc_quick_deploy** utility.

You can similarly define a Local infrastructure environment in a Quick Deploy script like the following example. Note the **infrastructureType** property that denotes the environment as a Local infrastructure. **imported** properties denote components imported from a Global infrastructure environment.

```
<?xml version="l.0" encoding="UTF-8" standalone="yes"?>
<quickDeployConfig configName="Env_001" version="l.0">
    <archType types=J2EE"/>
    <quickDeploySoftware>
        <software id="Foundation" version="2312.20231030.00"/>
    </quickDeploySoftware>
    <quickDeployApplications/>
    <quickDeployClients>
        <client id="fnd0_4tierrichclient" globalEnvName="GlobalEnvTestl5"
globalMachineName="testingl"
    globalInstallationPath="C:\Program Files\Siemens\Teamcenter\teamcenter_root">
            <property id="fnd0_4tierrichclient.infrastructureType" value="Local"/>
            <property id="fnd0_4tierrichclient.imported" value="true"/>
            <package id="fndO_foundation"/>
        </client>
        <client id="fndO_tccs" machineName="testingl" massDeploy="false"
platform="wntx64">
            <property id="fndO_tccsMergeFromFMSHorae" value=""/>
            <property id="fndO_tccsFscAssignmentMode" value="clientmap"/>
            <property id="fndO_tccsFccOptions" value="installFCC"/>
            <property id="fndO_tccsEnableClientCommunicationConfig"
value=""/>
            <property id="fndO_tccs.massDeployCheck." value="false"/>
            <property id="fndO_tccsInstallationPath" value="C:\Program
Files\Siemens\Teamcenter\teamcenter_root"/>
            <property id="fnd0_tccslocal" value="false"/>
            <property id="fndO_reverseProxyCriteriaList" value=""/>
            <property id="fnd0 tccsinfrastructureType" value="Local"/>
        </client>
    <quickDeployComponents>
        <component id="fnd0_corporateserver" machineName="testingl"
platform="wntx64
        <component id="fnd0_licensingserver" machineName="testingl"
platform="wntx64">
        <component id="fnd0_fsc" machineName="testingl"
platform="wntx64">
        <component id="fnd0_fsc keys" machineName="fsc"
platform="wntx64">
```

```
        <component id="fnd0_fsc group" machineName="fsc"
platform="wntx64">
        <component id="fnd0_tcdbserver" machineName="testingl"
platform="wntx64">
        <component id="fnd0_serverManager" machineName="testingl"
platform="wntx64">
        <component id="fnd0_j2ee tcwebtier" machineName="testingl"
platform="wntx64">
        <component id="fnd0_serverpool DBConfig" machineName="testingl"
platform="wntx64">
    </quickDeployComponents>
    <quickDeployProperties>
        <property id="InstallDirectory" value="C:\Program
Files\Siemens\Teamcenter\teamcenter root"/>
    </quickDeployProperties>
</quickDeployConfig>
```

Again, you can deploy your Local infrastructure environment using the **dc_quick_deploy** utility.

Troubleshoot quick deployment

Review the logs generated by the **dc_quick_deploy utility** on the Deployment Center server in *deployment_center_server_dir\logs*.

7. How to deploy the Business Modeler IDE templates on Teamcenter

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.