opentext™

OpenText™ Documentum™ Content Management

Transformation Services Installation Guide

Install and configure the content transformation services applications.

EDCCT250400-IGD-EN-01

OpenText™ Documentum™ Content Management
Transformation Services Installation Guide
EDCCT250400-IGD-EN-01
Rev.: 2025-Nov-27

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Installing OpenText Documentum Content Management (CM) Transformation Services

Transformation Services is a suite of Server products that perform transformations and analysis on repository content. Transformation Services functionality is available through Documentum client applications. It consists of the following products:

- OpenText Documentum Content Management (CM) Transformation Services - Documents
- OpenText Documentum Content Management (CM) Transformation Services - Media
- OpenText Documentum Content Management (CM) Transformation Services - Audio/Video
- OpenText Documentum Content Management (CM) XML Transformation Services

This guide is intended for a system administrator or a system operator. It is assumed that the system operator has a basic understanding of the Windows operating system.

This chapter describes the process of installing the Transformation Services products on a Windows host. Transformation Services runs as a Windows Service on the Transformation Services host.

Documentum post-installation tasks, configuration tasks, and troubleshooting information are also provided in this document.

Transformation Services allows for a high level of scalability. You can configure multiple Transformation Services instances to a single repository. If there is one installation of Transformation Services on a host, you can configure the same against different repositories.

Note: Clean up the previous versions of OpenText Documentum Content Management (CM) Foundation Java API before installing Transformation Services products.

1.1 Preinstallation configuration

This section describes all the pre-installation configuration that must be performed before installing Transformation Services.

1.1.1 Software prerequisites

This section provides information about the software that is required or optional for Transformation Services.

Server must be installed, configured, and it must be running on the network before installing Transformation Services.

Before installing Transformation Services, install several applications and other components on the server host (or another machine, where applicable). "Common requirements for Transformation Services" on page 6 provides details about the software that must be installed, configured, and running on the host (or another machine, where applicable) before installing Transformation Services.

Table 1-1: Common requirements for Transformation Services

Supporting software/ rendering applications	Requirement	Notes
Documentum Composer/ DAR installer	Required	To install DAR files. Need not be installed on the Transformation Services host, but must be able to access Transformation Services repositories.
Supported version of JDK installer	Required	Supported version of JDK must be installed on the Transformation Services system. The product *Release Notes* on OpenText My Support (support.opentext.com) contains information about the supported versions. Make sure that all environment variables are set before installing Transformation Services.
Microsoft Windows as the operating system	Required	
Documentum Administrator	Optional	To access the Transformation Services administration features.

Supporting software/ rendering applications	Requirement	Notes
Ghostscript **Note:** The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.	Optional	Allows the Doc3 plug-in to process PostScript files. Allows the Image3 plug-in to process all PostScript source and target formats such as EPS, PS, AI, and PDF.
Microsoft PowerPoint	Optional	Required by PowerPoint2 plug-in for the generation of thumbnails, low-resolution renditions, storyboards, and slide objects from the PowerPoint files.
Arial Unicode Font	Optional	To support extended multi-byte character range in documents (Chinese, Japanese, Korean, Arabic, and so on).
Microsoft HTML Help Workshop	Optional	Used for transforming XML files to CHM. This application must be installed with all options.
Microsoft Visual C++ 2022 Redistributable (64-bit) package	Optional	This Visual C++ Redistributable package must be installed for the Image3 plug-in to function.
Microsoft Visual C++ 2010 Redistributable (64-bit) package **Note:** The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.	Required	This Visual C++ 2010 Redistributable package must be installed to configure Transformation Services successfully.

Note: Make sure that all applications are installed in their default file paths. Transformation Services refers to each application's default installation directory while launching an application.

Some applications provide the choice of installing them with minimal or full options. Applications installed with minimal options may not run correctly with the Transformation Services product.

1.1.2 Selecting the server host

Follow these guidelines for selecting the host machine for Transformation Services installation.

- The server name must not exceed 15 bytes.
- The server must have 6 GB or higher System RAM. The Transformation Services server requires a 4 GB RAM for file manipulations. In addition, if the expected file size to be processed is large, then an equivalent memory allowance must be provisioned. For example, if the largest file expected is 1 GB, then the Transformation Services server requires an extra 1 GB RAM (4 + 1 GB). "Enhancing memory availability for Transformation Services" on page 29 provides instructions on how to increase the RAM for Transformation Services.

Caution

To ensure optimal system performance, do not install the Transformation Services product on the same host as Documentum CM Server or xPlore. This configuration is not certified, supported, or recommended.

1.1.3 Downloading installers

Before beginning the installation process, OpenText recommends you to download the installers on the Transformation Services host. Download the installers from My Support.

- cts-documents_<*release-version*>_windows.zip – Transformation Services - Documents
- cts-media_<*release-version*>_windows.zip – Transformation Services - Media
- cts-audio-video_<*release-version*>_windows.zip – Transformation Services - Audio/Video
- xts_<*release-version*>_windows.zip – XML Transformation Services

The ZIP file contains the executable installer file, silent installer batch files, and configuration files. Extract the ZIP files into a temporary folder before proceeding with the installation.

1.1.4 Verifying the connection broker

Verify the following before beginning the installation:

- If there is `dfc.properties` file, ensure that it is pointing to the correct connection broker. "Updating a connection broker in dfc.properties" on page 9 provides the instructions.
- The connection broker and repository services are running on the Documentum CM Server.

1.1.4.1 Updating a connection broker in dfc.properties

If there is OpenText™ Documentum™ Content Management Foundation Java API already installed on the Transformation Services host server, a `dfc.properties` file will be present. This procedure is to ensure that the correct connection broker is specified. If Transformation Services is installed on a clean host, a `dfc.properties` file does not exist before installation. In such a scenario, the following procedure is not required.

For more information about the connection broker, see *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)*.

To update a connection broker in dfc.properties:

1. Search for the `dfc.properties` file on the Transformation Services host server. The file is usually in the Windows (`C:\Documentum\config\`) folder.
2. Open the `dfc.properties` in a text editor.
3. Find the line indicating the connection broker. Ensure that the specified connection broker is the one that contacts the repository. Change it, if necessary.
4. Save and close the `dfc.properties` file.

1.1.4.2 Checking connection broker and repository services

The connection broker and repository services must be running correctly on the Documentum Server host before installing Transformation Services.

1.1.5 Setting the Java environment variables

Set the value of the Java environment variables as follows:

- *<JAVA_HOME>*: File path where Java is installed
- *path*: *<file path where Java is installed>*\bin\

1.1.6 Setting the locale of the repository and host

The locale setting of the repository and the Transformation Services host server must be the same before installing Transformation Services. If it is not, the `cts_instance_info` table is not created in the repository.

For example, install Transformation Services on a server where the locale is set to `en` and configure Transformation Services for a repository with locale set to `en`.

1.1.7 Installing Microsoft fonts

Fonts used in source files such as Microsoft Word, Excel, or PowerPoint must be available in the Transformation Services server host to get the correct target PDF renditions. Otherwise, the original fonts are replaced with alternate fonts in the resultant PDF rendition.

Note: The Arial Unicode MS font must be installed to support double-byte characters.

The Microsoft website provides the instructions required to install and uninstall Microsoft fonts.

1.1.8 Required installation information

The following table allows you to record the required information to install a Transformation Services server:

Table 1-2: Required installation information

Installation information	Description	Your value
OpenText Documentum Content Management (CM) Foundation Java API installation directory	This is the destination directory for Foundation Java API. The default directory is `C:\Documentum\`.	

Installation information	Description	Your value
Connection broker host name (if not detected)	If the correct Foundation Java API is not yet installed on the host, a prompt appears during installation to provide information for the primary connection broker host name and connection broker port number that is servicing the repository to configure with Transformation Services.	
Transformation Services server installation directory	The directory in which Transformation Services server is to be installed. OpenText recommends you to use a subdirectory of the directory in which other Documentum products are installed. For example, if other Documentum products are installed in `D:\Documentum\`, use the same path for Transformation Services installation. The installer must create a subdirectory called `CTS` as `D:\Documentum\CTS`.	
Transformation Services server host (if not detected)	The name of the machine on which the Transformation Services server is installed. Enter this information if it is not automatically detected by the installer, or if the installer is running from a remote computer.	
Administrator user name and password	The name of the administrator user and the password who is installing the Transformation Services server.	
Repository superuser name and password	The account the Transformation Services server uses to log in to the repository. Ask the administrator to provide the superuser name and password.	

Installation information	Description	Your value
Notification user name	The name of the user who needs to receive notifications from the Transformation Services server. Transformation Services notifies the user of the results of processing. For example, if the Transformation Services server fails to process an item, it queues an event to the Inbox of the Transformation Services server system operator user (`sysOpUser`).	
Transformation Services administrator port	The port used by Documentum Administrator to communicate with the Transformation Services server. The default port is `9095`.	
Transformation Services WebServices port	The port is used for WebServices communication. The default port is `9096`.	
Domain name	The name of the domain used to authenticate the user. Use the fully-qualified domain.	
Global registry	Different Foundation Java API installations can use different global registries where required service-based objects or network locations are stored. A single Foundation Java API installation can have only one global registry. A prompt appears during the installation for the primary registry.	
dm_bof_registry	By default, the global registry user has the user name of `dm_bof_registry`. During the repository configuration, provide the user login name and password for the global registry user.	

1.1.9 Providing required permissions to Transformation Services install user

Ensure that the Transformation Services install user has the following permissions:

1. The install user has administrator privileges.
2. The install user is part of the Windows local administrator group.
3. Add the install user to the **Log on as a service** policy in **Local Security Policy > Local Policies > User Rights Assignment**.
4. The install user has full permissions to the installation directory.

1.2 Installing Transformation Services

Ensure that the installation user account has administrator privilege and that the installation user account is included in the local administrators group. Otherwise, the Transformation Services product installation will fail.

Note: If Transformation Services is reinstalled, you must uninstall the previous version of the product that was installed on the server. For more information, see "Removing configuration from Transformation Services" on page 33.

To install Transformation Services:

1. Install the Transformation Services DAR files. For more information, see "Installing Transformation Services DAR files" on page 14.
2. Install Transformation Services.

   Install Transformation Services using one of the following installers:

   - Installer. For more information, see "Installing the Transformation Services products" on page 17.
   - Silent Installer. For more information, see "Installing and configuring Transformation Services products in silent mode" on page 25.
3. Configure the Transformation Services server to at least one repository. For more information, see "Configuring an instance of a Transformation Services server" on page 22.
4. After installing the Transformation Services server, verify if the installation is running correctly. For more information, see "Verifying Transformation Services installation" on page 31.

1.2.1 Installing Transformation Services DAR files

Ensure to install Transformation Services DAR files before configuring Transformation Services to obtain all the latest changes or enhancements.

Install the following DAR files in the global repository and in each repository where Transformation Services must be configured:

- CTSAspects.dar
- Rich_Media_Services.dar
- Transformation.dar
- XTS.dar

The DAR files can be installed from any machine with the latest version of headless Composer and the DAR installer plug-in on the local machine. The headless Composer is installed as part of Documentum CM Server. Download and install the latest version of headless Composer before installing the Transformation Services DAR files.

The headless Composer is available on My Support (support.opentext.com) as part of Documentum Developer Studio.

The headless Composer can also be found on the Documentum CM Server host at `C:\Documentum\product\<release-version>\install\composer\ComposerHeadless\`.

Note: Superuser privileges are mandatory for the repositories in which the DAR files are being installed.

On Windows, disable User Account Control (UAC) and invoke the DAR installer using the **Run as administrator** option.

1.2.1.1 Configuring headless Composer

The headless Composer ZIP file has the following format: `DCTM_Headless_Composer_<platform>_<release-version>.zip`

To configure headless Composer:

1. Extract the headless Composer ZIP file to a directory on the local machine. The directory name must not contain any spaces.
2. Open the `dfc.properties` file at `..\ComposerHeadless\plugins\com.emc.ide.external.dfc_1.00\documentum.config\` with a text editor.
3. Add the Foundation Java API and connection broker information as follows:

```
dfc.docbroker.host[0]=[Connection broker IP address or host name]
```

4. Save the changes and close `dfc.properties`.

1.2.1.2 Installing a DAR file with DAR installer plug-in

Use the DAR installer plug-in to install a DAR file to a repository to avoid the use of interface in Composer. The DAR installer plug-in requires Composer or headless Composer to be installed but does not launch the full Composer IDE. The DAR installer plug-in is useful for decoupling the DAR files development from the DAR files installation. It is also useful if the DAR files installation is required as part of a deployment process.

When the DAR installer program is executed, it creates the following folders in the Composer installation directory:

- `darinstallerconfiguration`: Contains configuration files for the DAR installer program.
- `darinstallerlogs`: Default file path of the log files.
- `darinstallerworkspaces`: Workspaces that are created and used by the DAR installer program. The DAR installer program does not delete these workspaces automatically. Therefore, occasionally the directory needs to be cleaned up. The workspace directories are named in the following format: `darinstallerworkspaces/yyyy-mm-dd-hh-mm-ss`

The DAR installer requires to fill in certain values that are marked with an asterisk (*). All other fields are optional.

DAR files are installed one at a time.

To install the DAR files:

1. Download the DAR installer ZIP file from the same place where Composer is downloaded. The DAR installer plug-in is found in the Documentum Composer software download location on My Support.
2. Extract the DAR installer ZIP file to the root directory of Composer or the headless Composer installation directory.
3. Run `dardeployer.exe` to start the DAR installer plug-in. The `dardeployer.exe` file is in the Composer root directory or on the Documentum CM Server machine at `C:\Documentum\product\<release-version>\install\composer\ComposerHeadless\`.
4. In the **DAR Details** section, specify values for the fields.
5. In the **Docbroker Details** section, specify values for **Docbroker Host** and **Docbroker Port** and click **Connect**.
6. In the **Repository Details** section, specify values for the fields and click **Install** to install the DAR file to the repository.

   "Description of the DAR installer fields (Transformation Services DAR)" on page 16 contains the description of DAR installer fields to help in the DAR file installation process.

7. Repeat step 2 through step 5 for the next DAR file.

To view the log for the DAR installation, select the log file from the **Log File** list and click **Open**.

Table 1-3: Description of the DAR installer fields (Transformation Services DAR)

Parameter	Required	Description
DAR	Yes	The absolute file path to the `.DAR` file that needs to be installed. The file path cannot contain any I18N characters. If 118N characters are used, the installation fails.
Input File	No	The absolute file path to the install-based parameter file.
Local Folder	No	The absolute file path to the localized `.properties` files. To make the application available in other languages, localize the project data as labels, tabs, and descriptions.
Log File	No	The file to save the log to. If this is not specified, the file defaults to *<DAR>*`.log`.
Connection Broker Host	Yes	The address of the connection broker.
Connection Broker Port	Yes	The port of the connection broker.
Repository	Yes	The name of the repository where the DAR file is installed. Click **Connect** after providing the connection broker host and port information to retrieve the available repositories.
User Name	Yes	The login name for the repository.
Password	Yes	The password for logging into the repository.
Domain	No	The domain where the repository resides.

Note: On a non-Windows Documentum CM Server machine, use the following command to install the DAR file:

```
java -Ddar=<darflepath> -Dlogpath=./<filename.log> -Ddocbase=<docbase> -Duser=<username>
-Ddomain=  -cp $DM_HOME/install/composer/ComposerHeadless/startup.jar
org.eclipse.core.launcher.Main -data $DM_HOME/install/composer/workspace
-application org.eclipse.ant.core.antRunner -buildfile $DM_HOME/install/
composer/deploy.xml
```

1.2.2 Installing the Transformation Services products

1. Log in to the selected Transformation Services server host as an administrator user.

   Note: Ensure that no programs or applications are running on the host.

2. Go to the folder containing the extracted Transformation Services server product installation files.
3. Double-click the following file to install this version of the Transformation Services product:

   - `CTS-DocumentsWinSetup.exe` to install Transformation Services - Documents
   - `CTS-MediaWinSetup.exe` to install Transformation Services - Media
   - `CTS-Audio-VideoWinSetup.exe` to install Transformation Services - Audio/Video
   - `xtsWinSetup.exe` to install XML Transformation Services

   Note: Perform the installation of the Transformation Services products consecutively, as required.

4. Click **Next**.
5. Accept the license agreement and click **Next**. The default destination directory `(C:\Documentum\)` for the installation of all components (Transformation Services Framework, Transformation Services, and Foundation Java API) appears on the screen.
6. Change the destination directory, if required, and click **Next**. To accept the default value, click **Next**.

   Note: Ensure that the selected destination directory does not contain a space in its name.

   Foundation Java API is installed in the same file path as Transformation Services.

7. Browse to the supported version of JDK directory and click **Next**.

   The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.

8. The **Installation Summary** screen appears listing the applications that will be installed.

- If satisfied with the installation confirmation, click **Next**.
- To change the component of the installation, click **Previous** to go back through the installer and change the installation information.

9. Click **Done**.

1.2.2.1 Configuring Transformation Services server with HashiCorp Vault

Vault securely stores and manages the secrets used by Transformation Services. It ensures that sensitive information, such as passwords, keys, and values, is protected and accessible only to authorized users.

To configure the Transformation Services server to use HashiCorp Vault, you must store all secrets in Vault. For more information about storing secrets in Vault, see *HashiCorp Vault* documentation.

To configure Transformation Services with HashiCorp Vault:

1. Sign in to the Vault server and create the following entries:

Note: User must provide Secret and Key name for HashiCorp Vault.

Table 1-4: Secret and Key name

Secret name	Key name	Description
INSTALL_OWNER_PASSWORD	*<repository_name>*	Installation owner password. For example, `INSTALL_OWNER_PASSWORD/testrepo`.
INSTALL_OWNER_PASSWORD	*<cts_hostname>*	Transformation Services administrator password. For example, `INSTALL_OWNER_PASSWORD/testcts`.
DFC_SECURITY_SSL_TRUSTSTORE_PASSWORD	*<cts_hostname>*	Foundation Java API Security SSL Truststore Password. For example, `DFC_SECURITY_SSL_TRUSTSTORE_PASSWORD/testcts`.

2. Copy the `dsis.zip` file from where you extracted the Documentum CM Server installer to a temporary location.

3. In the Transformation Services machine, open the `application.properties` file and set the `dsis.dctm.token` parameter as `True`.

   Note: Use the correct authentication mode to connect to Vault. For example, use OAuth modes such as AppRole, Token, and so on instead of the account-based OAuth of the Kubernetes that does not support virtual machines.

4. Ensure that the Vault type is mentioned as HashiCorp.

5. Start the DSIS daemon agent. For more information, see *Documentum CM Server* chapter of the *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

6. On the **Start** menu, click **Settings** > **CTS Configuration Utility** and click **Next** to run the Transformation Services configurator.

7. In the configurator, select the **Enable Vault** check box and provide the following information:

   a. **DSIS URL**: Provide the URL in the following format:

   ```
   http://localhost:<port mentioned in application.properties>/dsis
   ```

   b. **DSIS Token**: Provide the <*dsis_token*> token value as available in the `application.properties` file. For example, `dsis.dctm.token=2189790714430011293`.

   Note: If you enabled the HashiCorp Vault configuration, the installer retrieves all the password information automatically from Vault. Ensure that you have stored all the required secrets as described in the table "Secret name" on page 19.

Table 1-5: Secret name

Secret name	Customized keys
System Administrator Password Key	System administrator password. For example, `INSTALL_OWNER_PASSWORD/windows-cts-vm`
Global Registry Password Key	Global registry password. For example, `DFC_GLOBALREGISTRY_PASSWORD/testenv`
Repository Superuser Password Key	Repository superuser password. For example, `INSTALL_OWNER_PASSWORD/testenv`

Secret name	Customized keys
SSL Truststore Password Key	SSL truststore password. For example, `DFC_SECURITY_SSL_TRUSTSTORE_PASSWORD/testenv` **Note:** SSL Truststore Password Key is not mandatory for non-SSL configuration.

8. Click **Next** and proceed with the Transformation Services configuration. For more information, see "Configuring an instance of a Transformation Services server" on page 22.

Notes

- When configuring Transformation Services, if you have enabled Vault, you will not be prompted to provide the password.
- After the Transformation Services server starts running, you can stop the DSIS daemon agent.

  To restart the DSIS daemon agent, configure the environment variables in the following paths:

  – DFC_PATH - *<Documentum_home>*`\config\dfc_properties`

  – DSIS_PATH - *<dsis_home_path>*

After the environment variables are configured, the DSIS daemon agent will automatically start after the system restarts.

1.2.2.2 Configuring Transformation Services server with Kubernetes native secrets

To enable Kubernetes native secrets for Transformation Services, create an EC2 instance and configure the security settings to access and read from AWS Secrets Manager using the appropriate AWS policy. For more information about storing secrets in Kubernetes native secrets, see the *Kubernetes* documentation.

Transformation Services configured with Documentum CM Server that uses Kubernetes native secrets is supported only on the Amazon Web Services cloud platform. If the Documentum CM Server is deployed with Kubernetes Vault on a non-AWS cloud platform, configure Transformation Services to run without Vault.

To configure Transformation Services with Kubernetes native secrets:

1. Sign in to the Kubernetes native secrets server and create the following entries:

Note: User must provide only Secret for Kubernetes native secrets, and the Secret name can be customized.

Table 1-6: Secret name

Secret	Description
System Administrator Password Key	Install owner password. For example, <*ctsInstallOwnerPassword*>
Global Registry Password Key	Global registry password. For example, <*globalRegistryPassword*>
Repository Superuser Password Key	Repository superuser password. For example, <*dmadminPassword*>
SSL Truststore Password Key	SSL truststore password. For example, <*sslTruststorePassword*>

2. Copy the `dsis.zip` file from where you extracted the Documentum CM Server installer to a temporary location.

   Before starting the DSIS demon agent, update the `application.properties` file with the following settings in the Transformation Services machine:

   Note: Ensure that the `secretConfigName` value matches the name you specified when you created the secret.

   DSIS vault type:

   ```
   dsis.dctm.vault_type=AWS_K8API
   ```

   Secret configuration name:

   ```
   dsis.dctm.secretConfigName=vault-secret-config
   ```

3. In the Transformation Services machine, open the `application.properties` file and set the `dsis.dctm.token` parameter as `True`.

4. Ensure that the Vault type is mentioned as Kubernetes native secrets.

5. Start the DSIS daemon agent.

6. On the **Start** menu, click **Settings** > **CTS Configuration Utility** and click **Next** to run the Transformation Services configurator.

7. Select the **Enable Vault** check box in the configurator and provide the following information:

   a. **DSIS URL**: Provide the URL in the following format:

   ```
   http://localhost:<port mentioned in application.properties>/dsis
   ```

   b. **DSIS Token**: Provide the <*dsis_token*> token value as available in the `application.properties` file. For example, `dsis.dctm.token=2189790714430011293`.

Note: If you enabled the Kubernetes native secrets configuration, the installer retrieves all the password information automatically from Kubernetes native secrets. Ensure that you have stored all the required secrets as described in the table "Secret name" on page 21.

c. **System Administrator Password Key**: *<ctsInstallOwnerPassword>*

d. **Global Repository Password Key**: *<globalRegistryPassword>*

e. **Repository Superuser Password Key**: *<dmadminPassword>*

f. **SSL Truststore Password Key**: *<sslTruststorePassword>*

Note: SSL Truststore Password Key is not mandatory for non-SSL configuration.

8. Click **Next** and proceed with the Transformation Services configuration. For more information, see "Configuring an instance of a Transformation Services server" on page 22.

Notes

- When configuring Transformation Services, if you have enabled Kubernetes native secrets, you will not be prompted to provide the password.
- After the Transformation Services server starts running, you can stop the DSIS daemon agent.

  To restart the DSIS daemon agent, configure the environment variables in the following paths:

  – DFC_PATH - *<Documentum_home>*`\config\dfc_properties`

  – DSIS_PATH - *<dsis_home_path>*

  After the environment variables are configured, the DSIS daemon agent will automatically start after the system restarts.

1.2.2.3 Configuring an instance of a Transformation Services server

To complete the Transformation Services installation, configure the instance to a repository.

Note: Multiple Transformation Services instances can be configured to a repository.

Caution

- All the instances deployed against a repository must share the same version (including patch version) to avoid behavior conflicts between versions.

- Ensure that the appropriate connection broker and repositories are running before continuing with the configuration.

To configure a Transformation Services instance to a repository:

1. On the **Start** menu, click **Settings** > **CTS Configuration Utility** and click **Next**.
2. Type the host name of the machine running the Content Transformation Framework. Correct the host name if it is invalid or append the domain to create a Fully Qualified Domain Name and click **Next**.
3. Type the port number to which the Transformation Services Administration Agent is to be configured. This port is used with Documentum Administrator to communicate with Transformation Services hosts. The default port number is `9095`. Modify the default port number and click **Next**.
4. Type the port number to which Transformation Services WebServices is to be configured. The default port number is `9096` and click **Next**.

   Note: The installer automatically stops the Content Transformation Service if it is already available on the host.
5. Type the host name and port number of the primary connection broker. The default port number is `1489`.

   Note: This page appears only when a Transformation Services product is configured for the first time on a machine.
6. Select **Use certificates**. Click **Browse** to locate a valid `dfc.keystore` file. Type the password for the `dfc.keystore` file.

   Note: This step is optional and is performed only when the Transformation Services instance is configured to a repository that is projected with a secured connection broker (certificate-based). Alternatively, you can select **Use Default Java TrustStore** when configuring Transformation Services to a secured Documentum CM Server.
7. Type the global repository name, login name, and its password and click **Next**.

   Note: This page appears only when a Transformation Services product is configured for the first time on a machine.
8. Select **Add a Content Transformation Services instance to a repository** and click **Next**.

   Note: If a Transformation Services product is already configured in this repository, an additional option, **Remove a Content Transformation Services instance from a repository** appears with the **Add a Content Transformation Services instance to a repository** option. If all the installed Transformation Services products are configured in this

repository, only the **Remove a Content Transformation Services instance from a repository** option appears on the page.

9. Provide the information for the repository in the **Repository Name and User Information** page:

   a. Select a repository to be served by the Transformation Services server.

   b. Type the superuser name that the Transformation Services server needs to use to access the repository.

   c. Type the password for the superuser account.

   Note: The password must be between 8 to 64 characters long with at least one special character. However, do not use the % special character in the password to avoid any error.

   d. Type the domain name, if applicable. This is an optional field; if installing from a local machine, this can be left blank.

   e. Type the name of the repository user who needs to receive notifications from the Transformation Services server, otherwise known as the system operator.

   f. Select the **Add Additional Domain Users** check box, if you want to support requests from additional domains. If there are multiple domain users on Documentum CM Server that transform content, select this option. The installer then creates an administrator user for each domain. Enter credentials for a new domain and its superuser in the required fields. Repeat this step to include additional domains, as required.

   g. To generate performance metric reports, select **Set as Performance Log Repository**.

10. Click **Next**. If it is not possible to continue, check with the repository administrator to verify that the users exist and if the password provided is the correct password.

11. On the **CTS Configuration Program** page, select the Transformation Services products (Transformation Services - Documents, Transformation Services - Audio/Video, Transformation Services - Media, XML Transformation Services) that must be configured to the repository.

    Note: Only those Documentum Transformation Services products that were already installed are listed on this page.

    If only one Transformation Services product is installed, then this page does not appear.

12. Click **Next**.

13. Type the system administrator user name and password. Click **Next**.

14. On the **Profile Customization Report** page, select the previous version of Transformation Services configured to the repository to generate a Profile Customization Report.

Note: The **Profile Customization Report** page appears only when the configurator is run for the first time after installing Transformation Services.

15. After selecting a version, select the Transformation Services product that was previously configured.
16. Click **Next**. The registry updates. The configurator migrates all old profiles automatically, if applicable.
17. A dialog box confirms the success of the Transformation Services server configuration. Click **Profile Customization Report** to open the report.

    Restart the host if prompted to do so.

    Click **Done**.

Note: The Profile Customization Report is saved at `%CTS_HOME%\migration\log\`.

1.2.3 Installing and configuring Transformation Services products in silent mode

The silent installer allows to provide all installation, repository, and global registry information in a configuration file, run the installer and configurator batch files after this.

To install and configure Transformation Services products in silent mode:

1. Sign in to the selected Transformation Services server host as an administrator.
2. Ensure that no programs or applications are running on the host.
3. Go to the folder containing the extracted Transformation Services server product installation files. For more information about where the files are located, see "Downloading installers" on page 8.
4. Locate the `cts_win_install.properties` file.
5. Configure the `cts_win_install.properties` file with relevant values using a text editor as follows:

```
## silent install response file

INSTALLER_UI=silent


####installation
CTS.INSTALLATION_DIR=C:\\Documentum
PATH_TO_JAVA=C:\\jdk-<supported-version>
```

6. Open the command prompt and go to the installer directory where the `cts_win_install.properties` file is available.
7. Run the installer silently from the command prompt, using the following command:

```
start /wait <PRODUCT>WinSetup.exe -f  cts_win_install.properties
```

For example, for Transformation Services - Media, the command is as follows:

```
start /wait CTS-MediaWinSetup.exe -f  cts_win_install.properties
```

8. Verify the installation logs created in the logs folder in the installer directory.
9. After the successful installation, go to the *<CTS_HOME>*\server_install\ folder. This folder must contain the following files:
   - cts_win_config_adding.properties
   - cts_win_config_removing.properties

   Note: The *<CTS_HOME>* directory refers to C:\Documentum\CTS\ unless changed during the installation process.
10. To configure Transformation Services to a repository, update the cts_win_config_adding.properties file with the following relevant values using a text editor:

```
#all

#Specify the Admin port and Jetty port at first time config
CTS.ADMIN_PORT=9095
CTS.JETTY_PORT=9096

# Please specify the config actions, "add" or "remove".
CTS.INSTALL_TYPE=add
# Please specify the host name for CTS machine
CTS.HOSTNAME=cts-84
# Please specify the repository name
CTS.DOCBASE=repo1
# Please specify the repository super user and password
CTS.DOCBASE_SUPERUSER=Administrator
CTS.NOT_ENCRYPTED_PASSWORD=Install@bj1
# Please specify the domain name (optional)
CTS.DOMAIN=
# Please specify the user who receive notifications from CTS
CTS.DOCBASE_USER=Administrator
# Please set value to "true" if adding additional domain users, otherwise set it to
"false"
CTS.ADDITIONAL_DOMAINS=false
# Please set value to "true" if selecting this repository as Logging Performance
repository, otherwise set it to "false"
CTS.IS_PERFORMANCE_LOG_REPO=false

# Please specify the products which you want to configed if you install multiple
products in same machine. If you config multiple products, use comma as delimiter.
# Config will add all products, if this one remain empty.The value can be CTS-
Media, CTS-Documents, CTS-Audio-Video, XTS or combination any of them.
CTS.MODULE_NAMES=

# Please specify the system administrator name and password
CTS.SYS_ADMIN_NAME=Administrator
CTS.SYS_ADMIN_PASS=Install@bj1

# Please specify the global repository information
DFC.GLOBAL_REGISTRIES=reop1
DFC.GR_USERNAME=dm_bof_registry
DFC.DOCBROKER_HOST=win102
DFC.DOCBROKER_PORT=1489
DFC.SECURE.GR_PASSWORD=global_repository_password
```

```
#Below 2 options are used to generate a report of profile customizations applied in
the previous installation of CTS.
#Use this report to assess the profile customizations you want to apply to the
current installation.
#Only supported for previously installed CTS versions  22.1, 22.2, 22.4, 23.2,
23.4, 24.2, 24.4 and 25.2
#XTS was supported in 7.3 and later
#The report by default is available in this location <CTS HOME>/migration/log/
index.html
#Please input the product version you have previously installed. If not 20.2,
20.4,21.2,21.4 and 22.1, please leave it blank
CTS.OLD_VERSION=
#Please input the previously installed product name, valid product names are: CTS-
Media,CTS-Documents,CTS-Audio-Video,XTS(only valid for version 7.3)
#If you have never installed CTS product, please leave it blank. If you have more
than one product installed, please separate product name by comma.
CTS.OLD_PRODUCTS=
# Please specify SSL certificate information. Set "USE_CERTIFICATES" to "true" if
you want to enable SSL non-anonymous communication.
USE_CERTIFICATES=false
DFC_SSL_TRUSTSTORE=C:\\dfc.keystore
DFC_SSL_TRUSTSTORE_PASSWORD=password
DFC_SSL_USE_EXISTING_TRUSTSTORE=false

# Please specify the information for BOC support
CTS.BOCS_SELECTED=
CTS.ALLOW_BOCS_TRANSFER=
CTS.PREFER_ACS_TRANSFER=
CTS.ALLOW_SURROGATE_TRANSFER=

CTS.DOMAIN_USER_LIST=

# Please set value to "YES" if you want CTS to process contents in the distributed
store component that is local to the Content Server nearest, otherwise set it to
"NO".
CTS.PROCESS_LOCAL_CONTENT_ONLY=


IS_VAULT_ENABLED=false
DSIS_URL=
DSIS_TOKEN=

##don't change below values
INSTALLER_UI=silent
```

11. Run the installer silently from the command prompt, using the following command:

```
start /wait ctsConfigurator.exe -f cts_win_config_adding.properties
```

12. Verify the configuration logs created in the `%CTS_HOME%\logs\configurator.log\` folder.

1.3 Post-installation task and optional configurations

This section describes the post-installation task and optional configurations.

1.3.1 Licensing OpenText Documentum CM

OpenText Documentum CM uses OpenText Directory Services (OTDS) to apply licenses for all the OpenText Documentum CM components. For more information about procuring the license file and configuring OTDS and license, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

1.3.2 Optional configurations

This section describes the optional configurations.

1.3.2.1 Enabling Foundation Java API installation to access global registry repository

This section describes the process to configure the Foundation Java API installation to access a global registry, if a global registry is not selected during the installation, or if the password for the global registry user has changed, or if the `dfc.properties` file is updated with a different global registry than initially configured.

The `dfc.properties` file contains properties that relate to accessing the global registry.

To enable the Foundation Java API installation:

1. On the Foundation Java API host, go to `DOCUMENTUM\config\`.
2. From a command prompt, run the following command to generate the encrypted form of the global registry user's password:

   ```
   java -cp dfc.jar com.documentum.fc.tools.RegistryPasswordUtils <password_of_user>
   ```

   where *<password_of_user>* is the global registry user's clear-text password. In step 4, type the encrypted form of this password in the `dfc.properties` file.

   Note: Before running this command, ensure that the `classpath` contains `dfc.jar` or run the command from a file path where `dfc.jar` is available. For example, `C:\Documentum\Shared\`.

3. Open the `dfc.properties` file in a text editor.
4. Modify the following attributes:

   ```
   dfc.bof.registry.repository=<global_registry_repository_name>
   dfc.bof.registry.username=<user_login_name>
   dfc.bof.registry.password=<encrypted_password_of_user>
   ```

where *<encrypted_password_of_user>* is the encrypted password that was generated in step 2.

5. Save the `dfc.properties` file.

1.3.2.2 Configuring the Transformation Services server for another repository

This section describes the process to add more than one Transformation Services server configurations to the system.

Run one installation of Transformation Services on a single host and configure it against multiple repositories.

The procedure for adding an instance of Transformation Services is same as the procedure for the initial configuration of the product for a repository. This procedure does not perform the base installation or install configuration files, which must be done only once. There is no need to rerun the entire installation procedure to apply Transformation Services to an additional repository.

For more information about the procedure to configure the product for another repository, see "Configuring an instance of a Transformation Services server" on page 22.

1.3.2.3 Enhancing memory availability for Transformation Services

This section describes the process to increase the memory availability of Transformation Services.

The Transformation Services memory can be scaled up to increase performance or to allow processing of large files (> 1 GB). The default memory required for Transformation Services is 4 GB, but the memory must be increased for the transformation of large files. For every 1 GB RAM added to a minimum 6 GB system RAM, the Transformation Services memory can be increased using the following steps:

To increase the Transformation Services memory:

1. Sign in to the Transformation Services server host as an administrator user.
2. Select **Windows > Administrative Tools > Services** to access the Windows Services Utility.
3. Stop Transformation Services.
4. Open the Windows registry using `regedit.exe`.
5. Go to `[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DocumentumCTS\Parameters\]`. Modify the value of *AppParameters*.
6. Increase the -Xmx4096m value by adding 1024 to 4096 for every 1 GB extra RAM added. For example, for 2 GB extra RAM added, add 2048 to 4096. Hence, the new value is -Xmx6144m.

7. Close the registry.
8. Restart Transformation Services using the Windows Services utility.

1.3.2.4 Starting Content Transformation Service Admin Agent service

This section describes the post-configuration steps to start the Transformation Services server Admin Agent service.

To start the Transformation Services server Admin Agent service:

1. Run `services.msc` from the command prompt.
2. Go to `Documentum CTS Admin Agent` and stop it.
3. Right-click `Documentum CTS Admin Agent` and select `Properties`.
4. On the **Log On** tab, select the **Local System account** check box.
5. Click **Apply** and click **OK**.
6. Start the service.
7. Verify the generated logs at: `C:\Documentum\CTS\AdminAgent\logs\`
8. Verify the default Admin Agent URL: `http://<cts_machine_ip>:9095/CTSAgent/CTSAgent?`

Chapter 2

Verifying Transformation Services installation

This chapter explains how to verify an installation of Transformation Services.

2.1 Verifying Transformation Services installation

This section describes the following two methods to verify that Transformation Services is successfully installed :

- Verify if the following Windows services have been started using `services.msc`:
  - Transformation Services
  - Documentum Content Transformation Monitor Services
  - Documentum Transformation Services Admin Agent
- Import a test file to the repository.

Both tests are recommended to ensure that the installation is working correctly.

2.1.1 Verifying startup of Transformation Services service

After the Transformation Services server is installed and configured for a repository, Transformation Services should start automatically.

To verify startup of service:

1. Login to the Transformation Services server host as an administrator user.
2. Select **Windows > Administrative Tools > Services** to access the Windows Services utility.
3. Start Transformation Services.
4. Go to the *<CTS_HOME>*`\logs\` folder and open the `CTS_log.txt` file. Ensure that there are no errors or exceptions in the log file. At the end of the log file, a message that reads `Product` *<CTS product>* `version` *<version number>* `is installed on repository:` *<repository name>* must appear.

Chapter 3

Removing configuration and uninstalling Transformation Services

To install Transformation Services, uninstall previous versions of Transformation Services components. You must remove all the configuration from Transformation Services before uninstalling it. If you are installing it for the first time, skip to "Installing OpenText Documentum Content Management (CM) Transformation Services" on page 5.

3.1 Removing configuration from Transformation Services

This section provides information to remove all configuration from Transformation Services manually or in silent mode.

3.1.1 Removing configuration of Transformation Services instance manually

This section describes how to manually remove an instance of Transformation Services from a repository. This procedure does not remove the configuration files and services associated with the Transformation Services server. "Uninstalling Transformation Services products" on page 35 describes the process to remove the Transformation Services server from a host.

To remove Transformation Services manually:

1. Go to Transformation Services configurator in **Start > Programs > CTS Configuration Utility**.
2. Click **Next**.
3. Accept the license agreement and click **Next**.
4. Select **Remove a Content Transformation Services instance from a repository**, and click **Next**.
5. Select the repository from which you want to remove the Transformation Services server. Type the superuser name, the superuser password, and the domain if any.
6. Click **Next**.
7. On the **CTS Configuration Program** page, select the Transformation Services products (Transformation Services - Documents, Transformation Services - Audio/Video, Transformation Services - Media, XML Transformation Services) that must be removed from the repository.

Note: This screen does not appear if only one Transformation Services product is configured to the repository.

8. Click **Next**.

   If you cannot continue, check with the repository administrator to verify that the user exists and if you have provided the correct password.

9. Click **Done** indicating the success of the Transformation Services server removal.

   This instance of the Transformation Services server is now removed from the repository.

3.1.2 Removing configuration of Transformation Services instance in silent mode

The silent unconfigurator allows you to provide installation, repository, and global registry information in a configuration file, after which you can run the configurator batch files.

To remove Transformation Services in silent mode:

1. Login to the selected Transformation Services server host as an administrator user.
2. Ensure that no programs or applications are running on the host.
3. Update the `cts_win_config_removing.properties` file with relevant values using a text editor:

```
#all

# Please specify the config actions, "add" or "remove".
CTS.INSTALL_TYPE=remove
# Please specify the host name for CTS machine
CTS.HOSTNAME=MTS76
# Please specify the repository name
CTS.DOCBASE=repo1
# Please specify the repository super user and password
CTS.DOCBASE_SUPERUSER=Administrator
CTS.NOT_ENCRYPTED_PASSWORD=Install@bj1
# Please specify the domain name (optional)
CTS.DOMAIN=

# Please specify the products which you want to remove if you install multiple
products in same machine. If you remove multiple products, use comma as delimiter.
# Config will remove all products, if this one remain empty. The value can be CTS-
Media, CTS-Documents, CTS-Audio-Video, XTS or combination any of them.
CTS.MODULE_NAMES=

IS_VAULT_ENABLED=false
DSIS_URL=
DSIS_TOKEN=

##don't change below values
INSTALLER_UI=silent
```

4. Open the command prompt and go to the *<CTS_HOME>*`\server_install\` directory where the `cts_win_config_removing.properties` file is available. Run the installer silently from that command prompt using the following command:

```
ctsConfigurator.exe -f cts_win_config_removing.properties
```

3.2 Uninstalling Transformation Services products

This section describes the steps to uninstall the previous and current version of Transformation Services products.

All instances of Transformation Services deployed against a repository must share the same version (including patch version) to avoid behavior conflicts between versions. Therefore, before you install a new version of Transformation Services, you must uninstall any previous version of Transformation Services or Transformation Services products.

To uninstall Transformation Services products:

1. On the host machine, select **Start > Settings > Control Panel** and double-click **Add/Remove Programs**.
2. Select **Content Transformation Services** and click **Change/Remove**.
3. In the **Change/Remove** dialog box, click **Next**.
4. On the **Uninstall Content Transformation Services** screen, click **Next**.
5. On the **CTS Uninstall Program** screen, select the Transformation Services products (Transformation Services - Documents, Transformation Services - Audio/Video, Transformation Services - Media, XML Transformation Services) that must be uninstalled, and click **Next**.
6. If you want to uninstall Foundation Java API, select **Uninstall DFC**, and click **Next**.

   A dialog box confirms that the selected Transformation Services product is uninstalled.
7. Restart the Transformation Services server.

Chapter 4

Upgrading Transformation Services

When you upgrade Transformation Services to a new version, it migrates all the profiles by versioning them. Therefore, if you make changes to the OOTB profiles or command line files, ensure that you manually verify and reapply the changes to the profiles or command line files of the new version. In addition, all custom profiles and command line files must be manually verified against the current Transformation Services version for compatibility.

Note: Ensure that you process or clear all the queue items before upgrading to Transformation Services 7.0 or later. This is because the existing queue contains a reference to the `profile_id` of the pre-7.0 profile. Upgrading to the 7.0 release or later changes the profile and `r_object_id`. Therefore, the `profile_id` of the existing queue items becomes invalid.

4.1 Upgrading Transformation Services

This section provides information to upgrade Transformation Services from previous versions.

To upgrade Transformation Services:

1. Remove all the configuration from previous version of Transformation Services from the repository.
2. Uninstall Transformation Services.
3. Back up and remove the old profiles from the repository. For more information, see "Backing up and removing old profiles" on page 38.
4. Install and configure Transformation Services products. For more information, see "Installing OpenText Documentum Content Management (CM) Transformation Services" on page 5.
5. Assess the Profile Customization Report and manually apply the relevant customizations to the profiles.

4.2 Backing up and removing old profiles

The following is the behavior of the Transformation Services Migration utility:

- The Migration utility backs up the repository profiles in the file path specified in the *<BackUpFolder>* tag of the `script.xml` in the *<CTS_HOME>*`\migration\config\` folder. The default file path of the backup folder is *<CTS_HOME>*`\migration\backup\`.
- If the current version of Transformation Services has a profile with the same name as an existing profile in the repository, then the repository profile is reversioned to overwrite it.
- Custom profiles are not affected by the Migration utility. These custom profiles may issue errors during Transformation Services startup after migration. You must modify these profiles manually to make them compatible with the current version of the Transformation Services product.

If you have customized any OOTB profiles or added any profiles on a repository that were previously configured with Transformation Services, it is recommended to manually back up all these profiles before installing this version of Transformation Services. Export a copy of all profiles to the local disk before making any changes to the product installations.

It is also recommended, but not mandatory, to remove old OOTB profiles from the repository before installing this version of Transformation Services. This makes sure that only the current profiles included with this version of Transformation Services are contained in the `Media Server` folder.

To back up and remove profiles from a previous installation:

1. On the repository where Transformation Services is configured, Go to `\System\Media Server\`.
2. Back up any customized profiles from the previous version by exporting the contents of the `Media Server` folder to a local disk.
3. Delete the contents of the `Media Server` folder (including all versions and descendents).

4.2.1 Viewing log file of migrated profiles

During configuration of Transformation Services, the Migration utility is run on the existing profiles. You can view the log file generated by the profile migration tool. The default file path of the log file of migrated profiles is *<CTS_HOME>*\migration\log\.

The migration log file shows the result of the migration process: migration_log_2007_292_1192813617312.txt

The migration report log shows a complete list of all profiles before they are upgraded: migration_report_2007_292_1192813563921.txt

4.3 Post-upgrade task

This section describes the post-upgrade task.

4.3.1 Licensing OpenText Documentum CM

OpenText Documentum CM uses OpenText Directory Services (OTDS) to apply licenses for all the OpenText Documentum CM components. For more information about procuring the license file and configuring OTDS and license, see *OpenText Documentum Content Management - On-Premises Upgrade and Migration Guide (EDCCS250400-UMD)*.

Chapter 5

Deploying Transformation Services WebServices

Transformation Services WebServices is based on OpenText Documentum Content Management (CM) Foundation SOAP API platform. It provides synchronous real-time transformation capabilities. Any client application based on Transformation Services WebServices may request transformations that result in files that are available directly to the client, with minimal turnaround time.

Transformation Services WebServices is deployed using EAR or WAR.

Notes

- Always ensure that the version of Transformation Services WebServices is the same as that of the Transformation Services core product, even though they are installed on separate hosts.
- While deploying WebServices, you must deploy Foundation SOAP API in the same application server. For more information about deploying Foundation SOAP API, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

5.1 Enabling Vault for WebServices

Vault integrates with WebServices to securely manage sensitive information such as passwords, keys, and values. Based on your environment, you can configure one of the following Vault types:

- HashiCorp Vault
- Kubernetes native secrets

5.1.1 Enabling HashiCorp Vault for WebServices

To enable HashiCorp Vault for WebServices, you must store all Secrets and Key name in vault. For more information about storing secrets in vault, see *HashiCorp Vault* documentation.

To enable HashiCorp Vault for WebServices:

1. Sign in to the HashiCorp Vault server.
2. Store Secrets and Key name information in the `<secret_name>/<key_name>` format.

   For example, `INSTALL_OWNER_PASSWORD/webserviceshostname`
3. Copy the `DSIS.zip` file from the Transformation Services machine to the WebServices machine.

4. Start the DSIS daemon agent and ensure that the DSIS is running. For more information, see *Documentum CM Server* chapter of the *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

   Note: Stop the DSIS daemon agent after the Transformation Services are running.

5. Access the DSIS using the following URLs:

   ```
   http://localhost:8200/dsis/checkstatus
   ```

   The response code of 200 for the preceding URL indicates that the DSIS daemon agent has been initialized.

5.1.2 Enabling Kubernetes native secrets for WebServices

To enable Kubernetes native secrets for WebServices, you must store all Secrets in vault. For more information about storing secrets in Vault, see *Kubernetes* documentation.

To enable Kubernetes native secrets for WebServices:

1. Sign in to the Kubernetes native secrete server.
2. Store the secret name in the `<webserviceshostname>` format and the value.

   For example, Secret name: `webserviceshostname`, Value: `Password@123`.

3. Copy the `DSIS.zip` file from the Transformation Services machine to the WebServices machine.
4. Start the DSIS daemon agent and ensure that the DSIS is running.

   Note: Stop the DSIS daemon agent after the Transformation Services are running.

5. Access the DSIS using the following URLs:

   ```
   http://localhost:8200/dsis/checkstatus
   ```

   The response code of 200 for the preceding URL indicates that the DSIS daemon agent has been initialized.

5.2 Deploying Transformation Services WebServices on Red Hat JBoss

Deploying Foundation REST API on the JBoss EAP server is similar to any other web application deployment on the JBoss server. Transformation Services WebServices is deployed using the `transformation.ear` file.

To deploy Transformation Services WebServices on Red Hat JBoss:

1. Download the JBoss EAP 8 Update 8 patch file named `jboss-eap-8.0.8-runtime-maven-repository.zip` from the Red Hat website. The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.

2. Extract the `jboss-eap-8.0.8-runtime-maven-repository.zip` file to a temporary location.

3. Apply the JBoss EAP 8 Update 8 patch.

   Windows

   For example, from *<JBoss EAP_Home>*`\bin`, run the following command as an administrator:

   ```
   jboss-eap-installation-manager.bat update perform
     --dir C:\jboss-eap-8.0
     --repositories mrrc::file:C:\jboss-eap-8.0.8.GA-runtime-maven-repository
   \maven-repository
     --offline
   ```

   Linux

   For example, from *<JBoss EAP_Home>*`/bin`, run the following command as an administrator:

   ```
   ./jboss-eap-installation-manager.bat update perform
   >   --dir ../../jboss-eap-8.0
   >   --repositories mrrc::file:C:\jboss-eap-8.0.8.GA-runtime-maven-repository
   \maven-repository
   >   --offline
   ```

4. Create a folder named `cts_ws` in the working directory. For example, `C:\Documentum\`.

   a. Create the following subfolders inside the `cts_ws` folder:

   - `\cache\`
   - `\config\`
   - `\logs\`

   b. Create a subfolder `pfile` in the `config` folder. For example, `\config\pfile\`.

5. Open a text editor and create the `preferences.xml` file with the following content and save it in the *<WorkingDir>*`\cts_ws\config\` folder.

Note: This file must contain relevant values for the repository name, domain name, user name, and so on.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceConfiguration ID="CTS Web Services">
        <PropertyList>
            <ServerProperty Key="Cache" Description="The Temporary Cache Directory"
Value="C:\Documentum\cts_ws\cache" />
            <ServerProperty Key="AllowDirectTransfer" Description="Allow Direct
File
Transfer From CTS Server to Client. Set it to false if there is a firewall
restriction" Value="true" />
            <ServerProperty Key="CTSWSPingInterval" Description="Interval (in
seconds)
used to specify how frequent the LB should ping its CTS instances for heart
rate." Value="180" />
            <ServerProperty Key="FailoverRetries" Description="Allow a number of
retries if a request fails while waiting on the HTTP response from CTS"
Value="1" />
            <ServerProperty Key="InstanceSelector" Description="Specify an
implementation class for instance selection"
Value="com.emc.documentum.cts.lb.workers.DefaultSelector" />
        <Repositories>
            <AekFilePath>C:\Documentum\cts_ws\config\aek.key</AekFilePath>
            <LoginContext DocbaseName="<docbase_name_here>">
                    <ServerProperty Key="domain" Value="<domain_name_here>" />
                    <ServerProperty Key="userName" Value="<user_name_here>" />
                    <ServerProperty Key="passwordFile"
Value="C:\Documentum\cts_ws\config\pfile\mspassword.txt" />
                    <ServerProperty Key="maxConnectionRetries" Value="10" />
            </LoginContext>
        </Repositories>
    </PropertyList>
</ServiceConfiguration>
```

6. Copy the `aek.key` file from the Transformation Services server machine, located at `C:\Program Files\Documentum\CTS\config\`, and paste it in `C:\Documentum\cts_ws\config\` on the Transformation Services WebServices or Red Hat JBoss host.

7. Copy the `mspassword.txt` file from the Transformation Services server machine, located at `C:\Program Files\Documentum\CTS\docbases\`*`<docbase_name>`*`\config\pfile\`, and paste it in `C:\Documentum\cts_ws\config\pfile\` on the Transformation Services WebServices or Red Hat JBoss host.

8. Extract the contents of the `transformation.ear` file to a folder named `transformation.ear`.

9. Copy and paste the extracted `transformation.ear` folder to the `deployments` folder of a Red Hat JBoss server configuration. For example, go to `C:\`*`<Red Hat JBoss version folder>`*`\standalone\deployments\` and create the `transformation.ear.dodeploy` file in the `deployments` folder.

   Note: There are multiple configurations available that can be done with Red Hat JBoss. For example, default, minimal, and all.

   a. Update the `dfc.properties` file in `APP-INF\classes\` in `transformation.ear` to provide the correct details for connection broker, global registry, and so on. For example, open this file at `C:\`*`<Red Hat JBoss version folder>`*`\standalone\deployments\transformation.ear\APP-INF\classes\`.

It appears as follows:

```
dfc.data.checkout_dir=${dfc.data.user_dir}/checkout
dfc.data.dir=C:\Documentum\data
dfc.data.export_dir=${dfc.data.user_dir}/export
dfc.data.user_dir=${dfc.data.dir}
dfc.exception.include_decoration=false
dfc.exception.include_id=false
dfc.globalregistry.password=YiYQgefRI63JTWDokb7HJQ==
dfc.globalregistry.repository=CSFPI33_GR
dfc.globalregistry.username=dm_bof_registry
dfc.search.ecis.enable=false
dfc.search.ecis.host=
dfc.search.ecis.port=
dfc.tokenstorage.dir=${dfc.data.user_dir}/apptoken
dfc.tokenstorage.enable=false
dfc.docbroker.host[0]=10.31.107.145
dfc.docbroker.port[0]=1489
```

b. Update the `log4j2.properties` file in `APP-INF\classes\` to point the correct file path for all log files.

```
rootLogger.level = WARN
rootLogger.appenderRefs = A1, F1
rootLogger.appenderRef.A1.ref = STDOUT
rootLogger.appenderRef.F1.ref = File
monitorInterval = 5

#------------------- CONSOLE --------------------------
appender.A1.type=Console
appender.A1.name=STDOUT
appender.A1.layout.type=PatternLayout
appender.A1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.A1.filter.threshold.type = ThresholdFilter
appender.A1.filter.threshold.level = ERROR

#------------------- FILE -------------------------
appender.F1.type=RollingFile
appender.F1.name=File
appender.F1.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.log
appender.F1.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.
%d{yyyy-MM-dd}-%i
appender.F1.layout.type=PatternLayout
appender.F1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F1.policies.type=Policies
appender.F1.policies.time.type=TimeBasedTriggeringPolicy
appender.F1.policies.time.interval=1
appender.F1.policies.time.modulate=true
appender.F1.policies.size.type=SizeBasedTriggeringPolicy
appender.F1.policies.size.size=10MB
appender.F1.strategy.type=DefaultRolloverStrategy
appender.F1.strategy.max=5

#------------------- FILE_TRACE -------------------------

appender.FILE_TRACE.type=RollingFile
appender.FILE_TRACE.name=WEBSERVICEAppender
appender.FILE_TRACE.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.log
appender.FILE_TRACE.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.%d{yyyy-MM-dd}-%i
appender.FILE_TRACE.layout.type=PatternLayout
appender.FILE_TRACE.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.FILE_TRACE.policies.type=Policies
appender.FILE_TRACE.policies.time.type=TimeBasedTriggeringPolicy
appender.FILE_TRACE.policies.time.interval=3
appender.FILE_TRACE.policies.time.modulate=true
appender.FILE_TRACE.policies.size.type=SizeBasedTriggeringPolicy
appender.FILE_TRACE.policies.size.size=10MB
appender.FILE_TRACE.strategy.type=DefaultRolloverStrategy
appender.FILE_TRACE.strategy.max=6
```

```
#------------------- CTSWS -------------------------

appender.F34.type=RollingFile
appender.F34.name=CTSWSAppender
appender.F34.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F34.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F34.layout.type=PatternLayout
appender.F34.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F34.policies.type=Policies
appender.F34.policies.time.type=TimeBasedTriggeringPolicy
appender.F34.policies.time.interval=3
appender.F34.policies.time.modulate=true
appender.F34.policies.size.type=SizeBasedTriggeringPolicy
appender.F34.policies.size.size=10MB
appender.F34.strategy.type=DefaultRolloverStrategy
appender.F34.strategy.max=6
logger.R34.name =com.documentum.services.cts.impl.transform
logger.R34.level = INFO
logger.R34.additivity = false
logger.R34.appenderRef.F34.ref = CTSWSAppender
#-------------------------------------

appender.F35.type=RollingFile
appender.F35.name=CTSWSAppender1
appender.F35.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F35.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F35.layout.type=PatternLayout
appender.F35.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F35.policies.type=Policies
appender.F35.policies.time.type=TimeBasedTriggeringPolicy
appender.F35.policies.time.interval=3
appender.F35.policies.time.modulate=true
appender.F35.policies.size.type=SizeBasedTriggeringPolicy
appender.F35.policies.size.size=10MB
appender.F35.strategy.type=DefaultRolloverStrategy
appender.F35.strategy.max=6
logger.R35.name =com.emc.documentum.cts
logger.R35.level = INFO
logger.R35.additivity = false
logger.R35.appenderRef.F35.ref = CTSWSAppender1
#---------------------------------------

appender.F36.type=RollingFile
appender.F36.name=CTSWSAppender2
appender.F36.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F36.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F36.layout.type=PatternLayout
appender.F36.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F36.policies.type=Policies
appender.F36.policies.time.type=TimeBasedTriggeringPolicy
appender.F36.policies.time.interval=3
appender.F36.policies.time.modulate=true
appender.F36.policies.size.type=SizeBasedTriggeringPolicy
appender.F36.policies.size.size=10MB
appender.F36.strategy.type=DefaultRolloverStrategy
appender.F36.strategy.max=6
logger.R36.name =com.emc.documentum.fs.services.transformation
logger.R36.level = INFO
logger.R36.additivity = false
logger.R36.appenderRef.F36.ref = CTSWSAppender2

#------------------- DFS -------------------------

appender.F37.type=RollingFile
appender.F37.name=DFSAppender
appender.F37.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.txt
appender.F37.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.
```

```
%d{yyyy-MM-dd}-%i
appender.F37
```

10. Copy the `transformation.properties` file from `APP-INF\classes\` and paste it in the Red Hat JBoss Server's `bin` directory. For example, go to `C:\<Red Hat JBoss| version folder>\bin\`. This path might be different if Red Hat JBoss is installed in a different file path.

11. Update the `transformation.properties` file with the correct path to the `preferences.xml` file:

```
#cts ws preferences config location
CTSWSConfig=C:/Documentum/cts_ws/config/preferences.xml
```

12. Start the Red Hat JBoss server using `standalone.bat` in the `C:\<Red Hat JBoss version folder>\bin\`.

13. Optional To deploy Transformation Services on IPv6, perform the steps described in "Deploying Transformation Services in an IPv6 environment" on page 63.

14. Restart the Red Hat JBoss server using the following command depending on your operating system and communication protocol:

    a. On IPv4:

    Windows:

    ```
    standalone.bat -b 0.0.0.0
    ```

    Linux:

    ```
    standalone.sh -b 0.0.0.0
    ```

    b. On IPv6:

    Windows:

    ```
    standalone.bat -b [<IPv6 address>]
    ```

    Linux:

    ```
    standalone.sh -b <IPv6 address>
    ```

15. Ensure that the `TransformationService` and `ProfileService` are accessible from the Red Hat JBoss host and the client machine is using the following URLs:

    - Red Hat JBoss machine (runs on port 8080 by default):

      `http://<localhost/ip>:<port>/services/transformation/TransformationService`

      `http://localhost:<port>/services/transformation/TransformationService`

    - Remote client machine:

      `http://<Red Hat JBoss machine host/ip>:<port>/services/transformation/TransformationService`

      `http://<Red Hat JBoss machine host/ip>:<port>/services/transformation/ProfileService`

16. Send real-time transformation requests using the preceding service URL.
17. Check the logs in the `C:\Documentum\cts-ws\logs\` folder for the complete status of the Transformation Services WebServices server.

Notes

- For the JDK 17 support, add the following parameters in the `standalone.bat` file:

  set “JAVA_OPTS=-Dprogram.name=%PROGNAME% %JAVA_OPTS%”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-opens=java.base/java.net=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-opens=java.base/java.lang.ref=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-opens=java.naming/com.sun.jndi.toolkit.url=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-exports=java.base/sun.security.provider=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-exports=java.base/sun.security.pkcs=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-exports=java.base/sun.security.x509=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-exports=java.base/sun.security.util=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-exports=java.base/sun.security.tools.keytool=ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-opens=java.xml.crypto/com.sun.org.apache.xml.internal.security
  =ALL-UNNAMED”
  set “JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS%
  --add-opens=java.base/java.lang=ALL-UNNAMED”

- To avoid any JBoss EAP log issues, add the following line to the existing contents of the `jboss-deployment-structure.xml` file:

```
<exclude-subsystems><subsystem name="logging"/></exclude-subsystems>
```

5.3 Deploying Transformation Services WebServices on WildFly

Transformation Services WebServices is deployed using the `transformation.ear` file.

To deploy Transformation Services WebServices on Wildfly

1. Download WildFly. The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.
2. Create a folder named `cts_ws` in the working directory. For example, `C:\Documentum\`.

a. Create the following subfolders inside the `cts_ws` folder:

- `\cache\`
- `\config\`
- `\logs\`

b. Create a subfolder `pfile` in the `config` folder. For example, `\config\pfile\`.

3. Open a text editor and create the `preferences.xml` file with the following content and save it in the *<WorkingDir>*`\cts_ws\config\` folder.

Note: The `preferences.xml` file must contain relevant values for the repository name, domain name, user name, and so on.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceConfiguration ID="CTS Web Services">
        <PropertyList>
            <ServerProperty Key="Cache" Description="The Temporary Cache Directory"
Value="C:\Documentum\cts_ws\cache" />
            <ServerProperty Key="AllowDirectTransfer" Description="Allow Direct
File
Transfer From CTS Server to Client. Set it to false if there is a firewall
restriction" Value="true" />
            <ServerProperty Key="CTSWSPingInterval" Description="Interval (in
seconds)
used to specify how frequent the LB should ping its CTS instances for heart
rate." Value="180" />
            <ServerProperty Key="FailoverRetries" Description="Allow a number of
retries if a request fails while waiting on the HTTP response from CTS"
Value="1" />
            <ServerProperty Key="InstanceSelector" Description="Specify an
implementation class for instance selection"
Value="com.emc.documentum.cts.lb.workers.DefaultSelector" />
        <Repositories>
            <AekFilePath>C:\Documentum\cts_ws\config\aek.key</AekFilePath>
            <LoginContext DocbaseName="<docbase_name_here>">
                    <ServerProperty Key="domain" Value="<domain_name_here>" />
                    <ServerProperty Key="userName" Value="<user_name_here>" />
                    <ServerProperty Key="passwordFile"
Value="C:\Documentum\cts_ws\config\pfile\mspassword.txt" />
                    <ServerProperty Key="maxConnectionRetries" Value="10" />
            </LoginContext>
        </Repositories>
    </PropertyList>
</ServiceConfiguration>
```

4. Copy the `aek.key` file from the Transformation Services server machine, located at `C:\Program Files\Documentum\CTS\config\`, and paste it in `C:\Documentum\cts_ws\config\` on the Transformation Services WebServices or WildFly host.

5. Copy the `mspassword.txt` file from the Transformation Services server machine, located at `C:\Program Files\Documentum\CTS\docbases\`*<docbase_name>*`\config\pfile\`, and paste it in `C:\Documentum\cts_ws\config\pfile\` on the Transformation Services WebServices or WildFly host.

6. Extract the contents of the `transformation.ear` file to a folder named `transformation.ear`.

7. Copy and paste the extracted `transformation.ear` folder to the `deployments` folder of a WildFly server configuration. For example, go to `C:\<WildFly version folder>\standalone\deployments\` and create the `transformation.ear.dodeploy` file in the `deployments` folder.

   Note: There are multiple configurations available with WildFly. For example, default, minimal, and all.

   a. Update the `dfc.properties` file in `APP-INF\classes\` in `transformation.ear` to provide the correct details for connection broker, global registry, and so on. For example, open the file in `C:\<WildFly version folder>\standalone\deployments\transformation.ear\APP-INF\classes\`.

   It appears as follows:

```
dfc.data.checkout_dir=${dfc.data.user_dir}/checkout
dfc.data.dir=C:\Documentum\data
dfc.data.export_dir=${dfc.data.user_dir}/export
dfc.data.user_dir=${dfc.data.dir}
dfc.exception.include_decoration=false
dfc.exception.include_id=false
dfc.globalregistry.password=YiYQgefRI63JTWDokb7HJQ==
dfc.globalregistry.repository=CSFPI33_GR
dfc.globalregistry.username=dm_bof_registry
dfc.search.ecis.enable=false
dfc.search.ecis.host=
dfc.search.ecis.port=
dfc.tokenstorage.dir=${dfc.data.user_dir}/apptoken
dfc.tokenstorage.enable=false
dfc.docbroker.host[0]=10.31.107.145
dfc.docbroker.port[0]=1489
```

   b. Update the `log4j2.properties` file in `APP-INF\classes\` to point the correct file path for all log files.

```
rootLogger.level = WARN
rootLogger.appenderRefs = A1, F1
rootLogger.appenderRef.A1.ref = STDOUT
rootLogger.appenderRef.F1.ref = File
monitorInterval = 5

#------------------- CONSOLE -------------------------
appender.A1.type=Console
appender.A1.name=STDOUT
appender.A1.layout.type=PatternLayout
appender.A1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.A1.filter.threshold.type = ThresholdFilter
appender.A1.filter.threshold.level = ERROR

#------------------- FILE -------------------------
appender.F1.type=RollingFile
appender.F1.name=File
appender.F1.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.log
appender.F1.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.
%d{yyyy-MM-dd}-%i
appender.F1.layout.type=PatternLayout
appender.F1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F1.policies.type=Policies
appender.F1.policies.time.type=TimeBasedTriggeringPolicy
appender.F1.policies.time.interval=1
appender.F1.policies.time.modulate=true
appender.F1.policies.size.type=SizeBasedTriggeringPolicy
appender.F1.policies.size.size=10MB
appender.F1.strategy.type=DefaultRolloverStrategy
appender.F1.strategy.max=5
```

```
#------------------- FILE_TRACE -------------------------

appender.FILE_TRACE.type=RollingFile
appender.FILE_TRACE.name=WEBSERVICEAppender
appender.FILE_TRACE.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.log
appender.FILE_TRACE.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.%d{yyyy-MM-dd}-%i
appender.FILE_TRACE.layout.type=PatternLayout
appender.FILE_TRACE.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.FILE_TRACE.policies.type=Policies
appender.FILE_TRACE.policies.time.type=TimeBasedTriggeringPolicy
appender.FILE_TRACE.policies.time.interval=3
appender.FILE_TRACE.policies.time.modulate=true
appender.FILE_TRACE.policies.size.type=SizeBasedTriggeringPolicy
appender.FILE_TRACE.policies.size.size=10MB
appender.FILE_TRACE.strategy.type=DefaultRolloverStrategy
appender.FILE_TRACE.strategy.max=6

#------------------- CTSWS -------------------------

appender.F34.type=RollingFile
appender.F34.name=CTSWSAppender
appender.F34.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F34.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F34.layout.type=PatternLayout
appender.F34.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F34.policies.type=Policies
appender.F34.policies.time.type=TimeBasedTriggeringPolicy
appender.F34.policies.time.interval=3
appender.F34.policies.time.modulate=true
appender.F34.policies.size.type=SizeBasedTriggeringPolicy
appender.F34.policies.size.size=10MB
appender.F34.strategy.type=DefaultRolloverStrategy
appender.F34.strategy.max=6
logger.R34.name =com.documentum.services.cts.impl.transform
logger.R34.level = INFO
logger.R34.additivity = false
logger.R34.appenderRef.F34.ref = CTSWSAppender
#-------------------------------------

appender.F35.type=RollingFile
appender.F35.name=CTSWSAppender1
appender.F35.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F35.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F35.layout.type=PatternLayout
appender.F35.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F35.policies.type=Policies
appender.F35.policies.time.type=TimeBasedTriggeringPolicy
appender.F35.policies.time.interval=3
appender.F35.policies.time.modulate=true
appender.F35.policies.size.type=SizeBasedTriggeringPolicy
appender.F35.policies.size.size=10MB
appender.F35.strategy.type=DefaultRolloverStrategy
appender.F35.strategy.max=6
logger.R35.name =com.emc.documentum.cts
logger.R35.level = INFO
logger.R35.additivity = false
logger.R35.appenderRef.F35.ref = CTSWSAppender1
#-----------------------------------------

appender.F36.type=RollingFile
appender.F36.name=CTSWSAppender2
appender.F36.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F36.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F36.layout.type=PatternLayout
appender.F36.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F36.policies.type=Policies
```

```
appender.F36.policies.time.type=TimeBasedTriggeringPolicy
appender.F36.policies.time.interval=3
appender.F36.policies.time.modulate=true
appender.F36.policies.size.type=SizeBasedTriggeringPolicy
appender.F36.policies.size.size=10MB
appender.F36.strategy.type=DefaultRolloverStrategy
appender.F36.strategy.max=6
logger.R36.name =com.emc.documentum.fs.services.transformation
logger.R36.level = INFO
logger.R36.additivity = false
logger.R36.appenderRef.F36.ref = CTSWSAppender2

#------------------- DFS -------------------------

appender.F37.type=RollingFile
appender.F37.name=DFSAppender
appender.F37.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.txt
appender.F37.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.
%d{yyyy-MM-dd}-%i
appender.F37
```

8. Copy the `transformation.properties` file from `APP-INF\classes\` and paste it in the WildFly Server's `bin` directory. For example, go to `C:\<WildFly version folder>\bin\`. This path might be different if WildFly is installed in a different file path.

9. Update the `transformation.properties` file with the correct path to the `preferences.xml` file:

```
#cts ws preferences config location
CTSWSConfig=C:/Documentum/cts_ws/config/preferences.xml
```

10. Start the WildFly server using `standalone.bat` in the `C:\<WildFly version folder>\bin\`.

11. Optional To deploy Transformation Services on IPv6, perform the steps described in "Deploying Transformation Services in an IPv6 environment" on page 63.

12. Restart the WildFly server using the following command depending on your operating system and communication protocol:

    a. On IPv4:

    Windows:

    ```
    standalone.bat -b 0.0.0.0
    ```

    Linux:

    ```
    standalone.sh -b 0.0.0.0
    ```

    b. On IPv6:

    Windows:

    ```
    standalone.bat -b [<IPv6 address>]
    ```

    Linux:

    ```
    standalone.sh -b <IPv6 address>
    ```

13. Ensure that the `TransformationService` and `ProfileService` are accessible from the WildFly host and the client machine is using the following URLs:

- WildFly machine (runs on port 8080 by default):

  `http://localhost:<port>/services/transformation/TransformationService`

  `http://localhost:<port>/services/transformation/ProfileService`

- Remote client machine:

  `http://<WildFly machine host/ip>:<port>/services/transformation/TransformationService`

  `http://<WildFly machine host/ip>:<port>/services/transformation/ProfileService`

14. Send real-time transformation requests using the preceding service URL.
15. Check the logs in the `C:\Documentum\cts-ws\logs\` folder for the complete status of the Transformation Services WebServices server.

Notes

- If the JDK version is 17, add the following parameters in the `standalone.bat` file:

```
set "JAVA_OPTS=-Dprogram.name=%PROGNAME% %JAVA_OPTS%"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.net=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.lang.ref=ALL-
UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.naming/
com.sun.jndi.toolkit.url=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.provider=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.pkcs=ALLUNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.x509=ALLUNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.util=ALLUNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.tools.keytool=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.xml.crypto/
com.sun.org.apache.xml.internal.security=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.lang=ALL-
UNNAMED"
```

5.4 Deploying Transformation Services WebServices on Apache Tomcat

1. Install Tomcat on a host. The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.

   Note: This procedure is specific to Windows. If the Tomcat is running on a non-Windows machine, update to the non-Windows specific path, wherever applicable.

2. Stop the Tomcat server.
3. Create a folder with the name `cts_ws` in the working directory. For example, `C:\Documentum\`.

a. Create the following subfolders inside the inside the `cts_ws` folder:

- `cache`
- `config`
- `logs`

b. Create a subfolder `pfile` in the `config` folder (such as `\config\pfile\`).

4. Open a text editor and create a `preferences.xml` file with the following content and save it in the *<WorkingDir>*`\cts_ws\config\` file.

Note: The `preferences.xml` file contain relevant values for the repository name, domain name, user name, and so on.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <ServiceConfiguration ID="CTS Web Services">
    <PropertyList>
          <ServerProperty Key="Cache" Description="The Temporary Cache Directory"
                Value="C:\Documentum\cts_ws\cache" />
          <ServerProperty Key="AllowDirectTransfer" Description="Allow Direct File
                Transfer from CTS Server to Client. Set it to false if there is a
firewall
                restriction" Value="true" />
          <ServerProperty Key="CTSWSPingInterval" Description="Interval (in
seconds)
                used to specify how frequent the LB should ping its CTS instances
for heart rate."
                Value="180" />
          <ServerProperty Key="FailoverRetries" Description="Allow a number of
retries
                if a request fails while waiting on the HTTP response from CTS"
Value="1" />
          <ServerProperty Key="InstanceSelector" Description="Specify an
implementation
                class for instance selection"
Value="com.emc.documentum.cts.lb.workers.
                DefaultSelector" />
          <Repositories>
        <AekFilePath>C:\Documentum\cts_ws\config\aek.key</AekFilePath>
        <LoginContext DocbaseName="<docbase_name_here>">
        <ServerProperty Key="domain" Value="<domain_name_here>" />
        <ServerProperty Key="userName" Value="<user_name_here>" />
        <ServerProperty Key="passwordFile" Value="C:\Documentum\cts_ws\cache
                pfile\mspassword.txt" />
        <ServerProperty Key="maxConnectionRetries" Value="10" />
        </LoginContext>
          </Repositories>
    </PropertyList>
</ServiceConfiguration>
```

5. Copy the `aek.key` file from the Transformation Services Server machine in `C:\Documentum\CTS\config\` and paste it in `C:\Documentum\cts_ws\config\` on the Transformation Services WebServices or Tomcat host.

6. Copy the `mspassword.txt` file from the Transformation Services Server machine in `C:\Documentum\CTS\docbases\`*<repository_name>*`\config\pfile\` and paste it in `C:\Documentum\cts_ws\config\pfile\` on the Transformation Services WebServices or Tomcat host.

7. Extract the contents of the `transformation.war` file to a folder named `transformation`.

8. Copy and paste the extracted transformation folder to the `webapps` folder of a Tomcat server configuration.

   a. Update the `dfc.properties` file in `WEB-INF\classes\` to provide the correct details for connection broker, global registry, and so on.

      It appears as follows:

```
dfc.data.checkout_dir=${dfc.data.user_dir}/checkout
dfc.data.dir=C:\Documentum\data
dfc.data.export_dir=${dfc.data.user_dir}/export
dfc.data.user_dir=${dfc.data.dir}
dfc.exception.include_decoration=false
dfc.exception.include_id=false
dfc.globalregistry.password=YiYQgefRI63JTWDokb7HJQ==
dfc.globalregistry.repository=CSFPI33_GR
dfc.globalregistry.username=dm_bof_registry
dfc.search.ecis.enable=false
dfc.search.ecis.host=
dfc.search.ecis.port=
dfc.tokenstorage.dir=${dfc.data.user_dir}/apptoken
dfc.tokenstorage.enable=false
dfc.docbroker.host[0]=10.31.107.145
dfc.docbroker.port[0]=1489
```

   b. Update the `log4j2.properties` file in `WEB-INF\classes` to point to the correct file path for all log files.

```
rootLogger.level = WARN
rootLogger.appenderRefs = A1, F1
rootLogger.appenderRef.A1.ref = STDOUT
rootLogger.appenderRef.F1.ref = File
monitorInterval = 5

#------------------- CONSOLE -------------------------
appender.A1.type=Console
appender.A1.name=STDOUT
appender.A1.layout.type=PatternLayout
appender.A1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.A1.filter.threshold.type = ThresholdFilter
appender.A1.filter.threshold.level = ERROR

#------------------- FILE -------------------------
appender.F1.type=RollingFile
appender.F1.name=File
appender.F1.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.log
appender.F1.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.
%d{yyyy-MM-dd}-%i
appender.F1.layout.type=PatternLayout
appender.F1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F1.policies.type=Policies
appender.F1.policies.time.type=TimeBasedTriggeringPolicy
appender.F1.policies.time.interval=1
appender.F1.policies.time.modulate=true
appender.F1.policies.size.type=SizeBasedTriggeringPolicy
appender.F1.policies.size.size=10MB
appender.F1.strategy.type=DefaultRolloverStrategy
appender.F1.strategy.max=5

#------------------- FILE_TRACE -------------------------

appender.FILE_TRACE.type=RollingFile
appender.FILE_TRACE.name=WEBSERVICEAppender
appender.FILE_TRACE.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.log
appender.FILE_TRACE.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.%d{yyyy-MM-dd}-%i
appender.FILE_TRACE.layout.type=PatternLayout
appender.FILE_TRACE.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
```

```
appender.FILE_TRACE.policies.type=Policies
appender.FILE_TRACE.policies.time.type=TimeBasedTriggeringPolicy
appender.FILE_TRACE.policies.time.interval=3
appender.FILE_TRACE.policies.time.modulate=true
appender.FILE_TRACE.policies.size.type=SizeBasedTriggeringPolicy
appender.FILE_TRACE.policies.size.size=10MB
appender.FILE_TRACE.strategy.type=DefaultRolloverStrategy
appender.FILE_TRACE.strategy.max=6

#------------------- CTSWS -------------------------

appender.F34.type=RollingFile
appender.F34.name=CTSWSAppender
appender.F34.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F34.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F34.layout.type=PatternLayout
appender.F34.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F34.policies.type=Policies
appender.F34.policies.time.type=TimeBasedTriggeringPolicy
appender.F34.policies.time.interval=3
appender.F34.policies.time.modulate=true
appender.F34.policies.size.type=SizeBasedTriggeringPolicy
appender.F34.policies.size.size=10MB
appender.F34.strategy.type=DefaultRolloverStrategy
appender.F34.strategy.max=6
logger.R34.name =com.documentum.services.cts.impl.transform
logger.R34.level = INFO
logger.R34.additivity = false
logger.R34.appenderRef.F34.ref = CTSWSAppender
#-------------------------------------

appender.F35.type=RollingFile
appender.F35.name=CTSWSAppender1
appender.F35.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F35.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F35.layout.type=PatternLayout
appender.F35.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F35.policies.type=Policies
appender.F35.policies.time.type=TimeBasedTriggeringPolicy
appender.F35.policies.time.interval=3
appender.F35.policies.time.modulate=true
appender.F35.policies.size.type=SizeBasedTriggeringPolicy
appender.F35.policies.size.size=10MB
appender.F35.strategy.type=DefaultRolloverStrategy
appender.F35.strategy.max=6
logger.R35.name =com.emc.documentum.cts
logger.R35.level = INFO
logger.R35.additivity = false
logger.R35.appenderRef.F35.ref = CTSWSAppender1
#-----------------------------------------

appender.F36.type=RollingFile
appender.F36.name=CTSWSAppender2
appender.F36.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F36.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F36.layout.type=PatternLayout
appender.F36.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F36.policies.type=Policies
appender.F36.policies.time.type=TimeBasedTriggeringPolicy
appender.F36.policies.time.interval=3
appender.F36.policies.time.modulate=true
appender.F36.policies.size.type=SizeBasedTriggeringPolicy
appender.F36.policies.size.size=10MB
appender.F36.strategy.type=DefaultRolloverStrategy
appender.F36.strategy.max=6
logger.R36.name =com.emc.documentum.fs.services.transformation
logger.R36.level = INFO
logger.R36.additivity = false
```

```
logger.R36.appenderRef.F36.ref = CTSWSAppender2

#------------------- DFS -------------------------

appender.F37.type=RollingFile
appender.F37.name=DFSAppender
appender.F37.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.txt
appender.F37.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.
%d{yyyy-MM-dd}-%i
appender
```

9. Update the `transformation.properties` file in `WEB-INF\classes\` with the correct path to the `preferences.xml` file.

   It appears as follows:

   ```
   #cts ws preferences config location
   CTSWSConfig=C:/Documentum/cts_ws/config/preferences.xml
   ```

10. Copy the `transformation.properties` file from `WEB-INF\classes\` and paste it in the `tomcat/bin/` folder.
11. Start the Tomcat server.
12. Ensure `TransformationService` and `ProfileService` are accessible from the Tomcat host and a client machine is using the following URLs:

    - Tomcat machine (runs on port 8080 by default):

      `http://localhost:<port>/transformation/services/transformation/TransformationService`

      `http://localhost:<port>/transformation/services/transformation/ProfileService`

    - Remote client machine:

      `http://<Tomcat host/ip>:<port>/transformation/services/transformation/TransformationService`

      `http://<Tomcat host/ip>:<port>/transformation/services/transformation/ProfileService`

13. Send real-time transformation requests using the preceding service URL.
14. Check the logs in the `C:\Documentum\cts-ws\logs\` folder for the complete status of the Transformation Services WebServices server.

Notes

- If the JDK version is 17, add the following parameters in the `catalina.bat` file:

  ```
  set "JAVA_OPTS=-Dprogram.name=%PROGNAME% %JAVA_OPTS%"
  set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.net=ALL-UNNAMED"
  set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/
  java.lang.ref=ALLUNNAMED"
  set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.naming/
  com.sun.jndi.toolkit.url=ALL-UNNAMED"
  set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
  sun.security.provider=ALL-UNNAMED"
  set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
  ```

```
sun.security.pkcs=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.x509=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.util=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.tools.keytool=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.xml.crypto/
com.sun.org.apache.xml.internal.security=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.lang=ALLUNNAMED"
```

- To deploy Transformation Services on IPv6, perform the steps described in "Deploying Transformation Services in an IPv6 environment" on page 63.

5.5 Deploying Transformation Services WebServices on IBM Liberty

To configure Java, specify the Java version to be used in environment variables in the Liberty application server machine. After the JAVA_HOME and the path are set, the <*yourserver*>`console.log` displays the Java version that has been used to start the server.

To create a Liberty web application server manually:

1. The Liberty profile supports multiple servers using the same installation. Each server is defined in its own directory, where the directory name is the server name.

   a. Open a command line and change the directory to the wlp/bin. For example, `C:\IBM\WebSphere\Liberty\bin\`.

   b. Run the `server create` *<server_name>* command to create a server. For example, `C:\IBM\WebSphere\Liberty\bin>server create servercts\`.

   c. If you want to start, stop, or check the status of the Liberty web application server, use the following example commands:

      To start: `C:\IBM\WebSphere\Liberty\bin>server start servercts\`

      To stop: `C:\IBM\WebSphere\Liberty\bin>server stop servercts\`

      To check the status: `C:\IBM\WebSphere\Liberty\bin>server status servercts\`

2. Stop the IBM server.

3. Create a folder with the name `cts_ws` in your working directory. For example, `C:\Documentum\`.

   a. Create the following subfolders inside the `cts_ws` folder:

      - \cache\
      - \config\
      - \logs\

   b. Create a subfolder `pfile` in the `config` folder such as `\config\pfile\`.

4. Open a text editor and create a `preferences.xml` file with the following content and save it in the *<WorkingDir>*`\cts_ws\config` folder.

   Note: The `preferences.xml` file contain relevant values for your repository name, domain name, user name, and so on.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <ServiceConfiguration ID="CTS Web Services">
 <PropertyList>
 <ServerProperty Key="Cache" Description="The Temporary Cache Directory"
 Value="C:\Documentum\cts_ws\cache" />
 <ServerProperty Key="AllowDirectTransfer" Description="Allow Direct File
 Transfer from CTS Server to Client. Set it to false if there is a
firewall
 restriction" Value="true" />
 <ServerProperty Key="CTSWSPingInterval" Description="Interval (in
seconds)
 used to specify how frequent the LB should ping its CTS instances
for heart rate."
 Value="180" />
 <ServerProperty Key="FailoverRetries" Description="Allow a number of
retries
 if a request fails while waiting on the HTTP response from CTS"
Value="1" />
 <ServerProperty Key="InstanceSelector" Description="Specify an
implementation
 class for instance selection"
Value="com.emc.documentum.cts.lb.workers.
 DefaultSelector" />
 <Repositories>
 <AekFilePath>C:\Documentum\cts_ws\config\aek.key</AekFilePath>
 <LoginContext DocbaseName="<docbase_name_here>">
 <ServerProperty Key="domain" Value="<domain_name_here>" />
 <ServerProperty Key="userName" Value="<user_name_here>" />
 <ServerProperty Key="passwordFile" Value="C:\Documentum\cts_ws\cache
 pfile\mspassword.txt" />
 <ServerProperty Key="maxConnectionRetries" Value="10" />
 </LoginContext>
 </Repositories>
 </PropertyList>
</ServiceConfiguration>
```

5. Copy the `aek.key` file from the Transformation Services Server machine in `C:\Documentum\CTS\config\` and paste it in `C:\Documentum\cts_ws\config\` on the Transformation Services WebServices or the Liberty host.

6. Copy the `mspassword.txt` file from the Transformation Services Server machine in `C:\Documentum\CTS\docbases\`*<repository name>*`\config\pfile\` and paste it in `C:\Documentum\cts_ws\config\pfile\` on the Transformation Services WebServices or the Liberty host.

7. Copy the `transforamtion.war` file into the directory that you created in the step 1.b. For example, `/usr/servers/servercts/dropins`.

8. Update the `dfc.properties` file in `WEB-INF\classes\` to correct the details for connection broker, global registry, and so on. For example, `\servercts\apps\expanded\`.

   It appears as follows:

```
dfc.data.checkout_dir=${dfc.data.user_dir}/checkout
dfc.data.dir=C:\Documentum\data
```

```
dfc.data.export_dir=${dfc.data.user_dir}/export
dfc.data.user_dir=${dfc.data.dir}
dfc.exception.include_decoration=false
dfc.exception.include_id=false
dfc.globalregistry.password=YiYQgefRI63JTWDokb7HJQ==
dfc.globalregistry.repository=CSFPI33_GR
dfc.globalregistry.username=dm_bof_registry
dfc.search.ecis.enable=false
dfc.search.ecis.host=
dfc.search.ecis.port=
dfc.tokenstorage.dir=${dfc.data.user_dir}/apptoken
dfc.tokenstorage.enable=false
dfc.docbroker.host[0]=10.31.107.145
dfc.docbroker.port[0]=1489
```

9. Update the `log4j2.properties` file in `WEB-INF\classes\` to point to the correct location for all log files.

```
rootLogger.level = WARN
rootLogger.appenderRefs = A1, F1
rootLogger.appenderRef.A1.ref = STDOUT
rootLogger.appenderRef.F1.ref = File
monitorInterval = 5
#------------------- CONSOLE --------------------------
appender.A1.type=Console
appender.A1.name=STDOUT
appender.A1.layout.type=PatternLayout
appender.A1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.A1.filter.threshold.type = ThresholdFilter
appender.A1.filter.threshold.level = ERROR
#------------------- FILE -------------------------
appender.F1.type=RollingFile
appender.F1.name=File
appender.F1.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.log
appender.F1.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/log4j.
%d{yyyy-MM-dd}-%i
appender.F1.layout.type=PatternLayout
appender.F1.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F1.policies.type=Policies
appender.F1.policies.time.type=TimeBasedTriggeringPolicy
appender.F1.policies.time.interval=1
appender.F1.policies.time.modulate=true
appender.F1.policies.size.type=SizeBasedTriggeringPolicy
appender.F1.policies.size.size=10MB
appender.F1.strategy.type=DefaultRolloverStrategy
appender.F1.strategy.max=5
#------------------- FILE_TRACE -------------------------
appender.FILE_TRACE.type=RollingFile
appender.FILE_TRACE.name=WEBSERVICEAppender
appender.FILE_TRACE.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.log
appender.FILE_TRACE.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/
trace.%d{yyyy-MM-dd}-%i
appender.FILE_TRACE.layout.type=PatternLayout
appender.FILE_TRACE.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.FILE_TRACE.policies.type=Policies
appender.FILE_TRACE.policies.time.type=TimeBasedTriggeringPolicy
appender.FILE_TRACE.policies.time.interval=3
appender.FILE_TRACE.policies.time.modulate=true
appender.FILE_TRACE.policies.size.type=SizeBasedTriggeringPolicy
appender.FILE_TRACE.policies.size.size=10MB
appender.FILE_TRACE.strategy.type=DefaultRolloverStrategy
appender.FILE_TRACE.strategy.max=6


#------------------- CTSWS -------------------------
appender.F34.type=RollingFile
appender.F34.name=CTSWSAppender
appender.F34.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F34.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
```

```
%d{yyyy-MM-dd}-%i
appender.F34.layout.type=PatternLayout
appender.F34.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F34.policies.type=Policies
appender.F34.policies.time.type=TimeBasedTriggeringPolicy
appender.F34.policies.time.interval=3
appender.F34.policies.time.modulate=true
appender.F34.policies.size.type=SizeBasedTriggeringPolicy
appender.F34.policies.size.size=10MB
appender.F34.strategy.type=DefaultRolloverStrategy
appender.F34.strategy.max=6
logger.R34.name =com.documentum.services.cts.impl.transform
logger.R34.level = INFO
logger.R34.additivity = false
logger.R34.appenderRef.F34.ref = CTSWSAppender
#-------------------------------------
appender.F35.type=RollingFile
appender.F35.name=CTSWSAppender1
appender.F35.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F35.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F35.layout.type=PatternLayout
appender.F35.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F35.policies.type=Policies
appender.F35.policies.time.type=TimeBasedTriggeringPolicy
appender.F35.policies.time.interval=3
appender.F35.policies.time.modulate=true
appender.F35.policies.size.type=SizeBasedTriggeringPolicy
appender.F35.policies.size.size=10MB
appender.F35.strategy.type=DefaultRolloverStrategy
appender.F35.strategy.max=6
logger.R35.name =com.emc.documentum.cts
logger.R35.level = INFO
logger.R35.additivity = false
logger.R35.appenderRef.F35.ref = CTSWSAppender1
#---------------------------------------
appender.F36.type=RollingFile
appender.F36.name=CTSWSAppender2
appender.F36.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.txt
appender.F36.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/CTSWS_log.
%d{yyyy-MM-dd}-%i
appender.F36.layout.type=PatternLayout
appender.F36.layout.pattern=%d{ABSOLUTE} %6p [%t] %c - %m%n
appender.F36.policies.type=Policies
appender.F36.policies.time.type=TimeBasedTriggeringPolicy
appender.F36.policies.time.interval=3
appender.F36.policies.time.modulate=true
appender.F36.policies.size.type=SizeBasedTriggeringPolicy
appender.F36.policies.size.size=10MB
appender.F36.strategy.type=DefaultRolloverStrategy
appender.F36.strategy.max=6
logger.R36.name =com.emc.documentum.fs.services.transformation
logger.R36.level = INFO
logger.R36.additivity = false
logger.R36.appenderRef.F36.ref = CTSWSAppender2
#------------------- DFS -------------------------
appender.F37.type=RollingFile
appender.F37.name=DFSAppender
appender.F37.fileName=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.txt
appender.F37.filePattern=$C(DFC,DFC_DATA_CANONICAL_DIR)/cts_ws/logs/DFS_log.
```

10. Update the `transformation.properties` file in `WEB-INF\classes` with the correct path to the `preferences.xml` file.

    It appears as follows:

```
#cts ws preferences config location
CTSWSConfig=C:/Documentum/cts_ws/config/preferences.xml
```

11. Copy the `transformation.properties` file from `WEN-INF/classes` and paste it in the `C:\Program Files\IBM\WebSphere\Liberty\bin\` folder.
12. Start the server.
13. Ensure TransformationService and ProfileService are accessible from the Liberty host and a client machine is using the following URLs:

    - Liberty machine (runs on port 8080 by default):

      `http://localhost:<port>/transformation/services/transformation/TransformationService`

      `http://localhost:<port>/transformation/services/transformation/ProfileService`

    - Remote client machine:

      `http://<Liberty host/ip>:<port>/transformation/services/transformation/TransformationService`

      `http://<Liberty host/ip>:<port>/transformation/services/transformation/ProfileService`

14. Send real-time transformation requests using the preceding service URL.
15. Check the logs in the `C:\Documentum\ctsws\logs\` folder for the complete status of the Transformation Services WebServices server.

Notes

- If the JDK version is 17, add the following parameters in the `standalone.bat` file:

```
set "JAVA_OPTS=-Dprogram.name=%PROGNAME% %JAVA_OPTS%"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/java.net=ALL-
UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/
java.lang.ref=ALLUNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.naming/
com.sun.jndi.toolkit.url=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.provider=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.pkcs=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.x509=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.util=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-exports java.base/
sun.security.tools.keytool=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.xml.crypto/
com.sun.org.apache.xml.internal.security=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% --add-opens=java.base/
java.lang=ALLUNNAMED"
```

- To deploy Transformation Services on IPv6, perform the steps described in "Deploying Transformation Services in an IPv6 environment" on page 63.
- Update the `server.xml` file with following details:

For example, `C:\Program Files\IBM\WebSphere\Liberty\usr\servers\servercts\`.

```
<httpEndpoint id="defaultHttpEndpoint"
host="<IBM Liberty server IP>"
httpPort="8080"
httpsPort="9443" />
```

5.6 Deploying Transformation Services in an IPv6 environment

1. Set the following Java options in all the web application server configuration files:

   ```
   -Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true
   ```

2. In the web application server configuration files, change all the IP addresses to [::1] if the IP address is defined as 127.0.0.1.

Chapter 6

Troubleshooting the installation

This chapter provides information on troubleshooting Transformation Services installation. *OpenText Documentum Content Management - Transformation Services Administration Guide (EDCCT250400-AGD)* provides additional information regarding items that can affect configuration and usability.

6.1 Troubleshooting tips for Transformation Services

6.1.1 Checking the installer log file

If you receive notification that the Transformation Services server installation has failed, check the installer log file. By default, the log file (`installer.log`) is in the Transformation Services server installation file path identified in the section "Installing the Transformation Services products" on page 17.

6.1.2 If cts_instance_info object is not created when the configurator is executed

If the cts_instance_info object is not created when the configurator is run, there will be a problem with locale settings. The following errors will appear:

```
[dm_query2_e_data_dict_error_for_attr_a_c]error:
"the following error(s) occurred processing an alter/create
statement for type cts_instance_info, attribute cts_version."

[dm_query2_e_invalid_locale_setting]error:  "alter/create type:
the session_locale setting of 'en_gb' in the sessionconfig object
is not a valid locale in the docbase. the alter/create is
disallowed because it contains locale sensitive information for
an invalid locale. please either set the session_locale to a
valid dd_locales value ('en'), or add a new locale entry to
dd_locales (dm_docbase_config)."
```

The locale setting for the Transformation Services server host and the repository must match. If, for example, the Transformation Services machine has a locale setting of UK (en_gb) and the repository has a locale setting of US (en_us), the cts_instance_info type is not created.

The solution is to change the locale of the Transformation Services host to match the locale of the repository, or add the configurator machine's locale to the repository. The locale setting can be changed using DQL.

6.1.3 Transformation DAR error occurs when upgrading from a previous version

As part of an upgrade from a previous version, the `Transformation.dar` installation can fail with an error as follows:

```
[ERROR] Unexpected error occurred during install

com.emc.ide.installer.PreInstallException: Error during pre installation of procedure

...

Caused by: com.emc.ide.external.dfc.procedurerunner.ProcedureRunnerException: Error: The
procedure '%1' did not complete successfully.

   at com.emc.ide.external.dfc.procedurerunner.ProcedureRunnerUtils.executeDmBasic
(ProcedureRunnerUtils.java:206)
```

The error message says that a repository procedure failed to run and does not provide any details about what exactly caused it to fail.

To manually clean up the repository of Transformation Services-related items:

1. Go to the `System` cabinet.
2. In the `Media Server` folder, back up any custom profiles.
3. Delete the `Media Server` folder (including all versions and subfolders).
4. Go to the `System\Applications\` folder.
5. Delete all files and subfolders in the `\System\ApplicationsTransformation\` folder.
6. Go to the `System\Modules\TBO\` folder.
7. Delete the following folders (if present):
   - `dm_cts_response`
   - `dm_media_profile`
   - `dm_transform_request`
   - `dmc_cts_reporting_config`
   - `dmc_cts_request_archiving`
8. Go to the `System\Modules\SBO\` folder and:
   - Delete all the folders that start with `com.documentum.services.cts…;`
   - Delete all the folders that start with `com.documentum.services.dam….`
9. Restart the repository and try to reinstall the `CTS DAR` files.

6.1.4 Deleting log files

Log files generated by Transformation Services are not automatically removed from the file system, but a new log file is created every day. Log files with a timestamp before the current date can be deleted manually, if required. They are in the *<CTS_HOME>*\logs\ path.

6.1.5 Third-party applications installed on the Transformation Services host should not be used by any other products or users

Do not use any application that is installed on the Transformation Services host for any other uses other than Transformation Services.

If these applications are being used for some purpose other than by the Transformation Services server, there is a possibility of errors or failures in processing.

6.1.6 Uninstalling or reinstalling Transformation Services

Whenever you uninstall Transformation Services, OpenText recommends you to back up and remove the \System\Media Server\ folder in the repository itself. In addition, if you reinstall and the \System\Media Server\ folder already exists, delete it and restart the Transformation Services server. This recreates and updates the file path in the repository.

6.1.7 Transformation Services configurator cannot be configured to the Japanese, Korean, or Simplified Chinese repository

The Transformation Services configurator on an English operating system server does not configure a Japanese, Korean, or Simplified Chinese repository. The following error message appears:

```
The session locale setting of 'en' in the session config object is not a
valid locale in the repository.
```

The problem is that the Japanese, Korean, or Simplified Chinese repository must be populated first with the English locale (data dictionary), then the repository must be configured again using Transformation Services configurator.

To populate the Japanese, Korean, or Simplified Chinese repository with the English locale, perform the following steps in an English operating system:

1. Share the bin folder in Japanese/Korean/Simplified Chinese Documentum CM Server.

   Ensure that the bin folder is accessible from English operating system.

2. Install Foundation Java API on the English operating system with Japanese/ Korean/Simplified Chinese repository details.

3. Open command prompt and run the following command:

```
\\<Documentum Server IP address>\bin\dmbasic -f \\<Documentum Server IP address>\bin
\
dd_populate.ebs -e LoadDataDictionary -- <Repository Name> <Docbase
Owner><Password>
 \\<Documentum Server IP address>\bin\data_dictionary_en.txt
```

When connecting to another repository, the locale of the machine that is making the connection must be known to the repository; either the local machine must have the locale of `ko` or the server must add `en` as an acceptable locale in its locale table.

Chapter 7

Installing the language pack

Transformation Services customers who want to implement language packs have to install the Transformation Services language pack. The Transformation Services language pack includes the transformation profiles and mapping XML file that are included with Transformation Services. “Installing OpenText Documentum Content Management (CM) Transformation Services“ on page 5 provides information to install the English Transformation Services on the Server operating system. The procedure for installing Transformation Services language pack includes:

- “Installing a Transformation Services language pack” on page 69
- “Enabling multilingual support in Transformation Services” on page 70

7.1 Installing a Transformation Services language pack

Transformation Services customers who have installed the English Transformation Services product have to install the Transformation Services language pack to use language packs.

To download the language pack and stop Transformation Services:

1. Ensure that you have performed all the necessary prerequisites, as outlined in “Preinstallation configuration” on page 6 for the English product.
2. Download the required language pack (`CTS_Language_Pack_<LanguageName>_<ext>_<EXT>_<release-version>.xxxx.xxxx.zip` where `<LanguageName>` is the language you want to download, and `<ext>` or `<EXT>` are the extensions for the language you require to download) from My Support (support.opentext.com) to a temporary file path on the Transformation Services host.
3. Extract the contents of `CTS_Language_Pack_<LanguageName>_<ext>_<EXT>_<release-version>.xxxx.xxxx.zip`. It contains the following folders and files:
   - `mappings`
   - `profiles`
   - `CTS_Language_Pack_Version_File_<LanguageName>.txt`
4. Stop all instances of Transformation Services connected to the repository that are being updated with the language pack.

To add localized transformation profiles:

1. Using an appropriate Documentum client, sign in as an administrator user to the repository that are being updated.
2. Create a folder named `Resources` in the `\System\Media Server\` repository folder.
3. In the `Resources` folder, create separate folders for each language (that is, fr_FR, de_DE, it_IT, ja_JP, ko_KR, es_ES, zh_CN, ru_RU, pt_BR, en_US).
4. Import the contents of the `CTS_Language_Pack_<LanguageName>_<ext>_<EXT>_<release-version>.xxxx.xxxx\profiles\` folder into the appropriately named `\System\Media Server\Resources\` folder.

   Note: Ensure that the profiles are imported into the repository with the XSL extension.
5. Create a folder named `mappings` in the `\System\Media Server\Resources\<LanguageCode_CountryCode>\` repository folder.
6. Import the `CTS_mappings.xml` file from `CTS_Language_Pack_<LanguageName>_<ext>_<EXT>_<release-version>.xxxx.xxxx\mappings\` folder into the appropriately named `\System\Media Server\Resources\<LanguageCode_CountryCode>\mappings\` folder created in step 5.

To restart Transformation Services:

1. Copy the `CTS_Language_Pack_Version_File_<LanguageName>.txt` from the temporary folder (created in the download procedure) to the Transformation Services root file path, for example `C:\Documentum\CTS\`.
2. Restart the instances of Transformation Services that are connected to the repository to which the language pack is installed.

Note: Before using Transformation Services language packs, complete "Enabling multilingual support in Transformation Services" on page 70.

7.2 Enabling multilingual support in Transformation Services

Multilingual support can be enabled for Transformation Services by installing the Arial Unicode MS font on each Transformation Services machine to display different languages correctly.

This procedure is performed through the Windows Control Panel. After the installation of the Arial Unicode MS font, any language should be displayed properly if the operating system supports the characters specific to that language. However, there are some drawbacks to the Arial Unicode MS font that are as follows:

- Italics is not available.
- Spacing and kerning are slightly different compared to the regular Arial font.
- Double-width diacritic characters are not displayed properly.