opentext™

OpenText™ Documentum™ Content Management

Client Installation Guide

Install, upgrade, and configure client-side applications and features, including document viewers and PDF Configuration. Includes troubleshooting and uninstallation information.

EDCCL250400-IGD-EN-01

OpenText™ Documentum™ Content Management
Client Installation Guide

EDCCL250400-IGD-EN-01
Rev.: 2025-Nov-20

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

Preface

OpenText Documentum Content Management (CM) client consists of two components:

- Configuration: The web-based application, known as *client configuration*, for administrators to use to configure settings such as automated content-handling processes and background settings for the user interfaces.
- User interfaces: The web-based application, known as *Classic View*, provides the ability to interact with content in one or more repositories. A *Smart View* user interface is also available and can be installed from the standard client installer and configured using client configuration. Also available is OpenText Documentum CM Mobile (Mobile) for Android and iOS, that is available as a downloadable app and through mobile browsers. Except where noted, this guide refers to Classic View.

i Architecture diagram

The following diagram illustrates the architecture of the client:

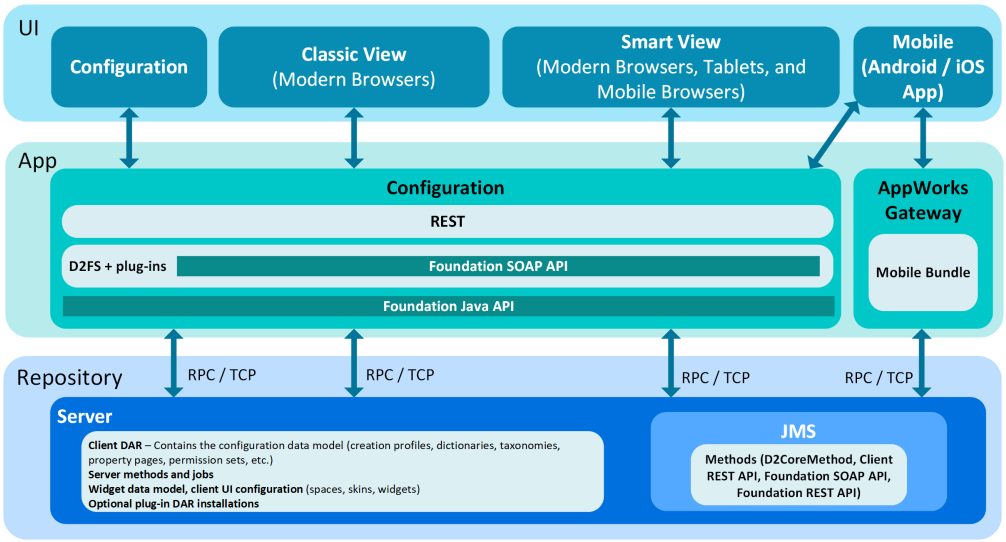

Figure 1: Client architecture

ii Intended audience

The information in this guide is for system administrators who install and administer the client.

Chapter 1

Get started

Know before you start

Before installing, make sure you know:

1. How to set the `CLASSPATH` environment variable.
2. The install paths for OpenText Documentum CM Documentum CM Server, Java Method Server (JMS), and your web application server.
3. How to set variable parameters for the Java Virtual Machine.
4. How to modify and deploy a `.war` package to your application server.
5. How to set variable parameters for your application server.

1.1 Prepare for installation

1. Read the *OpenText Documentum Content Management Release Notes* for your corresponding version for system requirements.
2. Make sure you have installed:
    1. Documentum CM Server, and have configured your repositories and docbrokers.
    2. A J2EE web application server as per your enterprise setup.
    3. Documentum Composer with a `DAR` installer for `.dar` deployment or Headless Composer for headless deployment of `.dar` files.

    Note: The client installer auto deploys the core `.dar` files. Documentum Composer with a `DAR` installer is required for manual deployment of `.dar` files. "Deploy DAR files manually" on page 51 contains further instructions on deploying `DAR` files manually.
3. Ensure you have administrator privileges on the local system to perform installation.
4. In a Linux environment, set the graphical environment, either by:
    - Adding the variable `java.awt.headless=true` to the environment system properties of the account running the application server.
    - Adding the parameter `-Djava.awt.headless=true` to the Java Method Server startup script.
    - The `$DOCUMENTUM` variable must be set for installer to run successfully. For `$DOCUMENTUM` variable please refer to the *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)*.

5. Download the following files to the Documentum CM Server and web application server machines:

File	Description	Content
Documentum Client *<version>*.zip	Contains the core installer; Classic SDK; Smart View Client; Smart View REST and SDK; Mobile Client; and the sample application to import after installation. **Notes** • If you are importing a separate application configuration, you do not need to import the provided sample. • When you install the D2-Doc-Library.dar file from d2-doclibrary-sampleapp.zip, change the value of the *dmadmin* parameter in the D2-Doc-Library.installparam file to [docbase owner account]: <parameter key="dmadmin" value=[docbase owner account]defaultValue="dmadmin" />.	• Documentum-Client-Installer-*<version>*.zip • Classic-SDK-*<version>*.zip • SV-SDK-*<version>*.zip • DocumentumMobile-*<version>*.zip • Documentum Client Sample Applications*<version>*.zip • D2-Config-Migrator-*<version>*.zip • D2-REST_*<version>*.war • Plugins folder
Documentum Client BOCS *<version>*.zip	Contains the Content Management Branch Office Caching Services for the OpenText Documentum CM client.	D2-BOCS-*<version>*.war

6. If you are upgrading, follow the instructions in "Plan your upgrades" on page 23.

   If you are upgrading from version 4.5 or earlier to version 21.4, you must run the Migration Utility before upgrading. To run the utility, follow the instructions in "Run the Migration utility" on page 24.

7. Follow the instructions for installing in "Instructions for installing the client" on page 14.

Important

Throughout the installation process, you may copy code from the documentation to paste into various files or scripts. Always ensure the pasted content perfectly reflects the information it was copied from, as characters, such as hyphens, are sometimes not copied over accurately.

1.2 Username restrictions

When you add usernames, keep in mind that some characters can prevent doclists from loading in the user interfaces. Examples of these characters can include:

- punctuation (,)
- symbols (@)
- German umlaut accent characters (ä ö ü)
- Spanish accent characters (á, é, í, ó, ú, ü, ñ)

If these characters are required in a username, add the following line to the `rest-api-runtime.properties` file:

```
rest.security.spring.httpfirewall.disable=true
```

The Spring security firewall by default will block some characters from the request URI that are valid when international characters are used for users. If Spring-blocked values appear in the request URI, Spring will reject the request and return "400 - Bad Request".

OpenText has excluded a few characters, meaning we are allowing fewer characters. These excluded characters are mentioned below:

```
FORWARDSLASH -> "//", "%2f", "%2F"
SEMICOLON -> ";", "%3b", "%3B"
BACKSLASH -> "\\", "%5c", "%5C"
ENCODED_PERIOD -> "%2e", "%2E"
ENCODED_PERCENT -> "%25"
```

These values could be coming as part of the genuine request URI so we have created a property called `rest.security.spring.httpfirewall.disable` in `rest-api-runtime.properties` which defaults to true. Based on the value, we are allowing this character also. The system further sanitizes requests using ESAPI for additional security.

1.3 About installation of Classic View and Smart View

The core installer includes two different web user interfaces for selection in the install wizard's **Select Installation Components** page. The traditional widget-based interface is identified as Classic View and the modern Smart View user interface is identified as Smart View.

If you are planning on performing an upgrade and want to include Smart View, select the **Customized Upgrade** option, which will allow you to include Smart View in the wizard's **Select Installation Components** page.

1.4 Instructions for installing the client

The following installation contains the steps for installing the Documentum Client API library files on the Documentum CM Server, deploying the DAR files, and then installing client configuration and the user interfaces on the web application server:

1. On the Documentum CM Servermachine:

   a. Follow the instructions in the "Install the client on Documentum CM Server" on page 41 section for installing the Documentum Client API libraries. Documentum Client API is a set of libraries for the Documentum CM Server and the JMS, enabling the client methods to be run on the Documentum CM Server.

   b. If you are installing on a Documentum CM Server cluster, install the Documentum Client API libraries on each instance of Documentum CM Server.

2. On the web application server:

   a. Stop web application server services.

   b. Follow the instructions for installing on the web application server as described in the following table:

Web application server	Instructions
Apache Tomcat	"Install on Apache Tomcat" on page 53
IBM WebSphere Liberty	"Install on IBM WebSphere Liberty" on page 58
Wildfly	"Install on Wildfly or Red Hat JBoss Server" on page 54

   c. Start application server services.

3. Configure:

The installer handles common Documentum CM Server and application server configuration options, these settings may be changed or additional optional settings may be configured post-installation.

a. Configure the Documentum CM Server as described in the following table:

Configuration	Instructions
Configuring `logback.xml`	"Configure logback.xml for Documentum CM Server" on page 76
Configuring the display of tables	"Configure Documentum CM Server server.ini" on page 77
Configuring auditing	"Configure auditing" on page 77

b. Configure the applications as described in the following table:

Application	Instructions
OpenText Documentum CM client configuration	"Configure client configuration" on page 65
User interfaces	"Configure the user interfaces" on page 68
Java Method Server	"Configure the Java Method Server (JMS)" on page 70

4. Configuring authentication protocols: "Configure authentication" on page 89
5. You can configure the file transfer mode. "File transfer modes" on page 70 contains more information about the WSCTF and thin client file transfer modes and "Configure file transfer modes" on page 75 contains configuration instructions.
6. Run the applications.
7. Log in to client configuration and import configurations. The *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)* contains further instructions on importing configurations.

   Using the client without configurations can cause errors. If you do not have a previous set of configurations to import, download and import `d2-base-starterapp.zip` as a sample set of configurations.

1.5 Set up an iJMS machine and configure to deploy iJMS

1. Create a new separate server for iJMS and add hosts file entries in both Documentum CM Server and iJMS server VM.
2. Install Java on the C: drive.
3. Install an iJMS build compatible with the Documentum CM Server version. Run `jmsStandaloneSetup.exe`. Choose the supported Java installation path as detailed in step 2.
4. Set the Documentum Server `dfc.properties` in the iJMS environment: `{installed Tomcat path}/webapps/DmMethods/WEB-INF/classes/dfc.properties`

   For example: `C:\JMS\tomcat10.1.29\webapps\DmMethods\WEB-INF\classes\dfc.properties`
5. Set the environment variable: `DM_JMS_HOME={installed Tomcat path}`

   For example: `DM_JMS_HOME=C:\JMS\tomcat<version>`
6. Copy `dfc.keystore` from the Documentum CM Server into the iJMS environment.

   For example: `C:\documentum\dba\secure\dfc.keystore`
7. Add the `dfc.keystore` path in the `C:\JMS\config\dfc.properties` file.

   Example:

   ```
   dfc.data.dir=C\:/JMS
   dfc.security.ssl.truststore=C:\\Documentum\\dba\\secure\\dfc.keystore
   dfc.security.ssl.truststore_password=AAAAEHZgq1Xs7EQCgCqRRoaObKpJRVx1dacO+wNlXe6RVunm
   ```
8. Run the `jmsStandaloneConfig` installer.
9. In the Documentum CM Server, restart the VM, the OpenText Documentum CM services, and also the iJMS service in iJMS VM.
10. Copy `Documentum-Client-Installer-<version>.zip` from the client build and extract `Documentum-Client-Installer-<version>.zip`.
    a. Double-click the **Documentum-Client-Installer** application to launch installer and then click **Next**.
    b. Select the **Install Documentum Client API for iJMS** check box and click **Install**.
    c. Select the **Documentum Client API for JMS** check box and click **Install**.

       Note: If BPM (Process Engine) is installed on the IJMS server VM , also select the **Documentum Client API for BPM** check box.
    d. Choose the required plug-ins and click **Next**.

e. The process will auto populate the required .jar files in `<installed Tomcat path>/webapps/DmMethods/WEB-INF/lib`.

Note: If the **Documentum Client API for BPM** check box is selected, the process will also autopopulate the required JAR files in `<installed Tomcat path>/webapps/bpm/WEB-INF/lib`.

f. Finish installation and click **Done**.

11. Restart the newly installed iJMS service.

1.6 Understand the Keystore utility

The Keystore utility replaces the functionality previously provided by the Lockbox utility.

Use the Keystore utility (`D2KeyStoreUtil`) to read from and write to the singleton d2_keystore object in the global registry repository.

Note: You must deploy `D2-DAR.dar` to the global registry repository before the Keystore utility can be run, or before any OpenText Documentum Content Management (CM) Foundation Java API client (such as client configuration or the user interfaces can be run.

By default, the keystore holds two encryption keys:

1. `transient_encryption_key`: used for certain transient operations. For example, encryption of admin login tickets for certain servlet URLs.
2. `encryption_key`: used to store external service passwords in encrypted form in the repository. For example, `d2_mail_config.push_password`, `d2_mail_config.pop_accept_password`, `d2_mail_config.pop_reject_password`

The values for these properties are automatically generated and there is no need to enter or modify these quantities using the Keystore utility unless it is specifically desired to change these for whatever reason. Note that if the value of the encryption_key property is changed, then one needs to re-enter the `d2_mail_config password` property values for each relevant repository using client configuration. See section 8.2 “Configuring the Mail Server” in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*.

The keystore is also used to hold various sensitive quantities such as repository admin login names and passwords that are needed for various optional features, including Single-Sign On (SSO) and LoadOnStartup initialization of dictionary caches, for example.

When the utility reads, it creates or overwrites the `d2keystore.properties` file in the current directory with the current contents of the keystore. When the utility writes, it clears the keystore in the global registry repository, reads the `d2keystore.properties` file in the current directory, writes these properties to the keystore in the global registry repository, and then deletes the `d2keystore.properties` file from the current directory.

Notes

- If the read command (`-r`) is run without running the write (`-w`) command, the `d2keystore.properties` file remains on the system. This file contains sensitive credential information and should be deleted manually if there are no plans to use the d2keystore utility to write information back into the d2keystore repository object.
- The global registry must be configured correctly with the proper password in `dfc.properties`.
- If the docbase ID is changed, you need to re-create `d2_keystore` by following these steps:
  1. Rename `d2_keystore` in the docbase from `DA System/D2` to `.keystore_old` (select **Show all objects and versions**).
  2. Create a keystore with the following command:

```
D2KeyStoreUtil.cmd -u dmadmin -p Password1! -c
```

1.7 Run the Keystore utility

Note: You must run the Keystore utility whenever a password changes for one of the administrative accounts that you have previously stored in the keystore.

To run the Keystore utility:

1. In a command shell, navigate to the directory in which the client configuration web application has been deployed, and then to the `d2keystore` subdirectory.
   - On Windows, this might be: `C:\Tomcat\webapps\D2-Config\utils\d2keystore`
   - On Linux, this might be: `/opt/Tomcat/webapps/D2-Config/utils/d2keystore`
2. Run the following command to read the current keystore properties and write them to a `d2keystore.properties` file.

   Note: The credentials (*<superUserLoginName>* and *<password>*) you provide to the `D2KeyStoreUtil` utility must be those of a `superuser` account in the global registry repository.

   - On Windows: `.\D2KeyStoreUtil.cmd -u` *<superUserLoginName>* `-p` *<password>* `-r`
   - On Linux: `./D2KeyStoreUtil.sh -u` *<superUserLoginName>* `-p` *<password>* `-r`
3. Edit the file `d2keystore.properties` and add appropriate lines for the properties that you wish to set in the keystore.

- If you are configuring **SSO using Foundation Java API principal mode authentication**, add the following two lines to specify admin credentials for all repositories of interest:

```
D2FS-trust.*.username=<username>
D2FS-trust.*.password=<password>
```

If some repositories have different admin credentials, you would add the following lines, where *repo1* and *repo2* are the names of the relevant repositories which require different admin credentials:

```
D2FS-trust.repo1.username=<repo1_username>
D2FS-trust.repo1.password=<repo1_password>
D2FS-trust.repo2.username=<repo2_username>
D2FS-trust.repo2.password=<repo2_password>
```

- If you are configuring **app server for LoadOnStartup initialization**, add the following two lines to specify admin credentials for all repositories of interest:

```
LoadOnStartup.*.username=<username>
LoadOnStartup.*.password=<password>
```

To specify admin credentials for all repositories of interest, or for repositories *repo1* and *repo2*, that require different admin credentials:

```
LoadOnStartup.repo1.username=<repo1_username>
LoadOnStartup.repo1.password=<repo1_password>
LoadOnStartup.repo2.username=<repo2_username>
LoadOnStartup.repo2.password=<repo2_password>
```

Note: Workflow reporting features might not operate as expected for completed or aborted Workflows if the `LoadOnStartup` setting is not enabled.

4. Run the following command to read the values from `d2keystore.properties` and write them to the keystore:

- On Windows: `.\D2KeyStoreUtil.cmd -u <superUserLoginName> -p <password> -w`
- On Linux: `./D2KeyStoreUtil.sh -u <superUserLoginName> -p <password> -w`

Notes

- If either the `transient_encryption_key` or `encryption_key` properties are not found in the `d2keystore.properties` file produced by running `D2KeyStoreUtil -r`, which you did in step 2, they will be automatically generated and written to the keystore when the utility is run again with the `-w` option, which you did in step 4.
- To get usage information for either of these scripts, run them without arguments. The current version of the underlying java command line utility outputs:

```
Usage: D2KeyStoreUtil -u superUserLoginName -p password [-r|-w]]

-r : Read d2_keystore properties from global registry repository and write
```

```
     to ./d2keystore.properties. The d2_keystore properties file will remain
on
     the system until the -w command is given or it is manually removed.

-w : Write d2_keystore properties in ./d2keystore.properties to global
     registry repository. The d2_keystore properties file is automatically
     deleted when -w is used.

where -r is assumed if neither -r nor -w is passed on the command line
```

All command line argument switches are case-sensitive.

5. Optional If you want to run these scripts, and the underlying java command line utility, from a different directory:

   a. Copy the scripts and the `D2KeyStoreUtiljar` file to the different directory.

   b. Edit the *<webinfdir>* variable inside the relevant script accordingly.

      The value of the *<webinfdir>* variable must be the path name (relative or absolute) for the `WEB-INF` directory of a deployed D2-Config web application.

To change the designated global registry repository:

1. Note that only one repository can be designated as the global registry repository. If this designation changes, careful planning is required in order to migrate the Keystore accordingly.
2. Before changing the designation, export the Keystore to a `d2keystore.properties` file using the Keystore utility with the "`-r`" option. See step 2.
3. After changing the designation, import the `d2keystore.properties` file created in step 2 using the Keystore utility with the "`-w`" option. See step 4.

Note: If the global registry repository changes, the `dfc.globalregistry.repository` must be changed in the `dfc.properties` files for all relevant Foundation Java API clients, including the `dfc.properties` file used by the Keystore command line utility.

1.8 Register Foundation Java API clients as privileged clients

All Foundation Java API clients must be approved as privileged clients in each applicable repository. Failure to approve a Foundation Java API client as a privileged client will result in errors when that Foundation Java API client attempts to access the keystore in the global registry repository or attempts to run code in privileged mode.

Approval is granted using Documentum Administrator.

Note: If you are using Business Admin Actions (see section 13.42 "Configuring Business Admin Actions" in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*), it is required to add Smart View as a

privileged client. You can do this using Documentum Administrator (see steps 1 to 4 below) or via command line using `d2privilegedclient` (see section 41.1 "d2privilegedclient" in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*).

1. You can view the list of Foundation Java API clients for a given repository that can be approved as privileged clients in Documentum Administrator by navigating to **Administration > Client Rights Management > Privileged Clients**. See *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)*.

2. If the clients that you want to approve as a privileged client do not appear in the list, click **Manage Clients**. A list of all registered Foundation Java API clients known to the repository is displayed.

   Note: Note that a Foundation Java API client is registered the first time it connects to Documentum CM Server. You might need to run the client once (for example, run the client configuration web application) in order to register it. To help you locate clients in the list of registered clients, you can filter the list by Client Name by entering a prefix in the text box with help text "**Starts with**" and clicking the arrow button to the right.

3. Select the desired clients from the list on the left-hand side and insert them into the list on the right-hand side. Click **OK**.

4. Right-click on a client and select **Approve Privilege**.

5. Optional With default settings in `dfc.properties`, it can be difficult to identify the registered clients that you want to approve in step 3 as privileged clients. To help, you can associate a client name with each by adding a `dfc.name` property to the client's `dfc.properties` file. For example: `dfc.name=D2_10.141.58.212`

   The Client Name is prefixed with this value when it appears in the list of Privileged Clients or list of registered clients on the **Manage Clients** dialog. You can then filter by Client Name to find registered clients that have a given prefix in their name.

6. Optional To confirm that the client you selected is the one that you want to approve as a privileged client in step 3, you might need to compare its Client ID value as displayed in Documentum Administrator with the client ID value stored in its `dfc.keystore` file.

   To find the client ID, use the java `keytool` command line utility:

   a. Open a command prompt and navigate to the folder where the `dfc.properties` file for your Foundation Java API client is located. Typically, this folder is the `WEB-INF/classes` folder of the associated web application. If the `dfc.security.keystore.file` property in the `dfc.properties` file has not been set, then this folder contains the desired `dfc.keystore` file. If it has been set, then the value of this property will give the path to the desired `dfc.keystore` file.

Note: The `dfc.keystore` file is created the first time a Foundation Java API client connects to Documentum CM Server.

b. Once you have located the folder for the `dfc.keystore` file, navigate to this folder in a command shell and run the java `keytool` command:

```
keytool -list -keystore dfc.keystore -storepass dfc -v
```

c. The resulting output contains detailed information about the certificate stored in the keystore. The value of the `CN` parameter in the `Owner` field is the desired Client ID. For example, if the resulting output contains the following line:

```
Owner: CN=dfc_t2zN2bJv8DDyMlZbXD9qJJTWiXAa, O=EMC, OU=Documentum
```

The desired Client ID is `dfc_t2zN2bJv8DDyMlZbXD9qJJTWiXAa`

Chapter 2

Upgrade the client on Documentum CM Server

Plan your upgrades

Pre-requisites:

Notes

- In the usual course of operation, it is no longer a requirement to delete preferences during upgrades or uninstall operations. Certain unusual scenarios might require this mitigation, so the option to delete preferences is still available.
- Configurations from version 3.x of the client cannot be imported into or exported from version 20.4 and higher.

1. Stop web application services.

   Note: The DocBroker services must be running.

2. Delete the cookies and cache of the web browsers.
3. Back up the previous WAR files. "Configuration files" on page 187 contains the locations of configuration files.
4. Delete the temporary installer files.

   For example, in Microsoft Windows, delete the folder `C:\Users\Administrator\AppData\Local\Temp\2\DCTMCM_<version>`.

   If you do not delete the temporary installer files, the installation may not overwrite property files.

   Notes

   - Make sure you *only delete* the *contents* of the `%USERPROFILE%\AppData\Local\Temp\<#>` folder and not the folder itself.

     The JVM API uses this folder to store the temporary files. If this folder is deleted, the JVM API will reference a non-existent location that will result into errors when running the core installer and configuration utility. If the `%USERPROFILE%\AppData\Local\Temp\<#>` folder is deleted, restart the server before running the core installer and configuration utility.
   - If you are using Foundation Java API older than 21.2 and `log4j2.properties` is present in the `\WEB-INF\classes` folder, rename it to `log4j2.properties.bak`.

In the same location, rename `log4j.properties.bak` to `log4j.properties`.

Note: If you are upgrading the client to version 25.2 on Documentum Content Server 24.4, apply Content Server Patch 24.4.1 first, then upgrade the client to 25.2.

Upgrade older versions to 24.4 or later

Upgrading from older versions to version 24.4 or later must include running `D2CoreMethod`. Configure the method with the below parameters to ensure all the objects are updated with the new ACL key.

For example:

```
-dql_filter select * from dm_document where i_cabinet_id='0c01767a80003ea8' -create
false, -security true, -apply_for_vd false, -force_security true
```

Note: Make sure `-dql_filter` is updated per the requirements of the environment.

2.1 Run the Migration utility

Note: Running the Migration utility is required only when you are using version 4.5 or earlier and planning to upgrade to version 21.4. If you are already using version 4.6 or later, the migration utility is not needed.

If you are planning to upgrade your version, check for the pre-requisites and upgrade tasks required to upgrade .

The Migration utility is a Foundation Java API standalone utility that is run on the Documentum CM Server machine to migrate the client configuration objects in the docbase.

The Migration utility must be run to ensure existing environments adhere to the new config object model, which now extends `dm_sysobject` and thereby grants appropriate ACLs to config objects. The utility is shipped as a standalone ZIP file along with the required libraries.

The Migration utility works in the following phases:

- *Prepare*: Creates session, authenticates user, validates environment, checks for installed plug-ins and creates folder for configs.
- *Dump*: Dumps the objects that are being migrated as xml files in the dump folder. Dump includes configs, types, as well as dm_relations objects. It also scans dumped xml files.
- *Upgrade*: Deletes objects and types in the docbase, and invokes Headless Composer to install the new types. The deletion of data cannot be reverted.
- *Create*: Recreates objects and relations based on the dump files. Updates any old object ID references in dm_relations.

- *Validate*: Prints migration statistics after the run.

Remember the following points before running the Migration utility:

- The Migration utility requires that `DM_HOME` environment variable is set on the machine. This is required to invoke the Headless Composer pre-packaged along with the docbase. This is set by default.
- "`Error retrieving object by Object Id`". This may happen if an object previously installed by Documentum Composer was deleted and `DM_TYPE_MGR_E_CANT_FIND_TYPE_HANDLE` in Documentum Composer logs will be encountered while running the migration utility. This does not impact the migration in any manner and can be safely ignored.
- Run the Migration utility from Documentum CM Server machine where migration is performed.
- In Windows Server, run command prompt as administrator if the windows logged in user does not have administrator privileges.
- The docbase user should be a `superuser`.
- Install Java 1.8 or later on the Documentum CM Server machine where the utility runs.
- Migration phases can be controlled by command line arguments.
- The migration utility can be re-run if migration fails at any point in time.
- The migration utility performs migration by doing dump, deletion, and recreation of data. OpenText strongly advises you to backup the Documentum CM Server and Database machines before performing migration in production environments. The process cannot be reverted except through backup restores or snapshot based solutions.
- In version 4.2, the **Dictionary** attribute in creation profiles was not mandatory, but it is mandatory starting in version 4.5. Set the **Dictionary** drop-down before migrating to avoid corrupting the `creation_profile`.
- To avoid possible data corruption issues, consult with OpenText Support before manually terminating the migration utility.
- Delete all `d2c_subscription_queue` and `c6_method_return` objects before running the migration utility. These objects are not essential and can be safely removed before migration to avoid issues during the migration process.

To run the Migration utility:

1. Extract `D2-Config-Migrator-21.4.zip` to the Documentum CM Server machine.
2. The `dfc.properties` file in the folder, by default, includes the line: `C:/Documentum/config/dfc.properties`

   For non-Windows machines, update the path accordingly.
3. Check Services:

a. Ensure that the Documentum CM Server and docbroker services are running.

b. Stop the web application server and JMS.

4. Open command prompt and run the utility using following command:

```
java -jar D2-Config-Migrator.jar <docbase> <loginname> <password>
```

Set the parameters for the command as described in the following tables. The first table details the mandatory parameters and the second table details the optional parameters. The optional arguments need not be used unless the migration fails repeatedly and an override is required for any of the phases.

Mandatory Parameter	Description
<docbase>	Name of the docbase repository.
<loginname>	Install owner login name.
<password>	Install owner password.

Optional Parameter	Description
`skipCreateConfigFolders`	Skips creating sub-folders for every config type under `/System/D2/Data` in repository.
`skipDumpConfigs`	Skips dumping D2-Config objects.
`skipScanDumpRelations`	Skips dumping `dm_relations` related to these configs.
`skipDumpTypes`	Skips dumping `dm_type` that are migrated.
`skipScanDumpedConfigs`	Skips scanning dumped config xml files to determine objects already migrated in previous run.
`skipScanDumpedRelations`	Skips scanning dumped relation xml files to determine relations already migrated in previous run.
`skipDeleteConfigs`	Skips deleting config objects from repository.
`skipDeleteTypes`	Skip dropping types from repository.
`skipInstallDars`	Skips Headless Composer dar upgradation.
`skipCreateConfigs`	Skips recreation of configs from dump files.
`skipCreateRelations`	Skips recreation of relations from dump files.
`skipUpdateRelationDescriptions`	Skips updating object id references in `dm_relation` objects.

Optional Parameter	Description
forceMigrate	Force migration on the repo. Dump config, dar installation, and delete types phases will be forced irrespective of arguments.
help	Lists out all possible arguments for the command.

5. The Migration utility does not register the table for a dictionary you selected for **Create Register Table**. In this case, follow these manual registration steps:

   a. In client configuration, navigate to the applicable dictionary and deselect **Register Table**.

   b. Save the dictionary, then select it again and save a second time. A register table will be created.

   c. To confirm the creation of the register table, run the following DQL query and review the result:

   ```
   select * from dbo.dictionaryname
   ```

6. Validate the migration.

   After running the utility, the folder where the utility was extracted will contain the following sub-folders and files:

   - `composer/`: Contains the dars used by Headless Composer during migration.
   - `dump/<docbase>/configs/`: Contains all config instances dumped.
   - `dump/<docbase>/relations/`: Contains all relation instances dumped.
   - `dump/<docbase>/types/`: Contains all config types dumped.
   - `lib/`: All libs packaged with the migration utility.
   - `logs/`: Contains `Migration.log`, `composer-out.log`, `composer-err.log` files.
   - `D2-Config-Migrator.jar`: The Migration utility jar.
   - `dfc.properties` file

2.2 Upgrade on Documentum CM Server

Before upgrading ensure that the following prerequisites are met:

- Navigate to the `C:\Documentum` folder where the `.D2InstallationInfo` file entries or path can be edited and pointed to the new path before the upgrade. If you do this, the upgrade will not replace `.jar` files, but will make a backup of the old `.jars` and replace with new ones.
- Ensure that there are no prior installations of the JDK in the `C:\Documentum\java64` directory. For example, if an older JDK is present in that directory, it will remain and the `.dar` installation may fail. Ensure only a supported JDK version is found here or that the directory is empty so that a supported JDK can be pointed to during install.

- If you are using WildFly or Tomcat to deploy, update the `java.policy` or `permissions.xml` file first. Update the JRE by editing or creating the `java.policy` file (`java64\JAVA_LINK\conf\security\java.policy`) with the following permission:

```
grant {
permission com.documentum.fc.client.impl.bof.security.RolePermission "*",
"propagate";
};
```

  If you are using Wildfly, add the `permissions.xml` in `ServerApps.ear\META-INF`, `BPM.ear\META-INF` on Documentum CM Server and the `META-INF` folder for the client application server to give propagate action for `RolePermission`:

```
<?xml version="1.0" encoding="UTF-8"?>
  <permissions xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/
javaee http://xmlns.jcp.org/xml/ns/javaee/ permissions_7.xsd" version="7">
    <permission>
      <class-name>com.documentum.fc.client.impl.bof.security.RolePermission</class-
name>
      <name>*</name>
      <actions>propagate</actions>
    </permission>
  </permissions>
```

Once version 23.4 or later has been installed, the details of the installation are saved in the `.D2InstallationInfo` file in the `<Documentum installation folder>`. For example: `C:\Documentum`

If you run the installer again to modify your previous installation settings the options to perform a express or custom upgrade is available:

1. Extract `Documentum-Client-Installer-<version>.zip`.
2. Double-click the**Documentum-Client-Installer** application to launch installer and then click **Next**.
3. Select **I accept the terms of the license agreement** to continue with the installation.
4. Select **Upgrade Documentum Client**.

   You get two upgrade options:

   - Express Upgrade
   - Customized Upgrade (includes selection for Smart View)

   Note: If applicable, you can click the **Merge Documentum Client customizations** check box to retain your existing plug-ins and customizations between patches and upgrades.

5. Select **Express Upgrade** to upgrade your installation using the previous installation settings. Click **Install** and perform the following.

   a. Review the summary page for all the parameters and pre-configured values. Click **Next**.

b. The **Smartview Configuration** panel appears. On this page, specify the crypto salt value for client token encryption and decryption. These values are updated in the `rest-api-runtime.properties` file. Clear the check box if you would like to remove the Swagger Documentation from `D2-Smartview/rest`. Click **Next**.

c. On the plug-ins installation panel, select plug-ins by checking the corresponding check boxes to install new plug-ins or update existing ones. Click **Next**.

   You can select **Core Signature**, **DocuSign**, **PDF Configuration**, **Transfer Configuration**, **Recycle Bin**, **Retention Policy Services (RPS)**, **InfoArchive**, **Extended ECM (xECM)**, and **Reports** plug-ins. The installer automatically runs the added plug-in installer files, activates the plug-in, and automatically deploys DAR files.

   For example, if you select **PDF Configuration**, you do not need to:

   - deploy the output `C2-API.jar` and `C2-Plugin.jar` files,
   - configure `D2-Config.properties`, or
   - deploy `C2-DAR.dar`.

   If you do not add plug-ins using the installer, you can manually run the plug-in installer `.bin` for Linux and `.exe` for Windows, and the plug-in installer automatically copies the JAR files to the respective folders.

   Notes

   - On selecting the **Reports** plug-in, the plug-in JARs (`DCTM-Reports-Plugin` and `DTR-API`) are copied, and the respective DARs (`DCTM_Reports_Base.dar`, `DCTM_Reports_GBL.dar`, and `DCTM_Reports_Template_for_D2.dar`) are deployed.
   - For more information on installing digital signature plug-ins, see "Digital signature components" on page 169.

d. Type the Documentum CM Server install owner's name and password. Confirm the password.

e. When you select the **Classic View**, **Client Config**, or **Smart View** pack, the **Client Language Pack** panel appears.

   Select the required localization by selecting the appropriate check boxes to install the corresponding locales. Click **Install**.

   Note: Install the locale to the repository. The *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* contains future instructions on populating and publishing localized data dictionaries to the repository.

   You can select **Arabic**, **German**, **Spanish**, **French**, **Italian**, **Japanese**, **Korean**, **Dutch**, **Portuguese**, **Swedish**, and **Chinese** locales. The installer automatically copies the locale JARs to the respective WARs.

For example, if you select **Arabic**, you do not need to:

- copy the locale JARs (`C2-LanguagePack_ar.jar`, `D2-Bin-languagePack_ar.jar`, `D2FS4DCTM-LanguagePack_ar.jar`, `D2-LanguagePack_ar.jar`, and help folder) to `tomcat\webapps\D2\WEB-INF` or `tomcat\webapps\D2\help` or
- copy the locale JARs (`D2FS4DCTM-LanguagePack_ar.jar` and UI folder) to `tomcat\webapps\D2-Smartview\WEB-INF` or `tomcat\webapps\D2-Smartview\ui`.

If **Client Config** and **French** are selected, you do not need to:

- copy the locale JARs (`C2-LanguagePack_fr.jar`, `D2-Bin-languagePack_fr.jar`, `D2FS4DCTM-LanguagePack_fr.jar`, `D2-LanguagePack_fr.jar`, and help folder) to `tomcat\webapps\D2\WEB-INF` or `tomcat\webapps\D2\help`,
- copy the locale JARs (`C2-LanguagePack_fr.jar`, `D2-Bin-languagePack_fr.jar`, `D2FS4DCTM-LanguagePack_fr.jar`, `D2-LanguagePack_fr.jar`, `D2-Config-LanguagePack_fr.jar`, `D2-InfoArchive-LanguagePack_fr.jar`, `D2-RPS-LanguagePack_fr.jar`, `D2-Specifications-LanguagePack_fr .jar`, `D2-xECM-LanguagePack_fr.jar`, `O2-LanguagePack_fr.jar`, and help folder) to `tomcat\webapps\D2-Config\WEB-INF` or `tomcat\webapps\D2-Config\help`, or
- copy the locale JARs (`D2FS4DCTM-LanguagePack_fr.jar`, `D2-InfoArchive-LanguagePack_fr.jar`, `D2-xECM-LanguagePack_fr.jar`, and UI folder) to `tomcat\webapps\D2-Smartview\WEB-INF` or `tomcat\webapps\D2-Smartview\ui`.

Note: Ensure that the following entry in the `rest-api-runtime.properties` file in `D2-Smartview\WEB-INF\classes` displays your intended locales, in short form, separated by commas (default is en). For example: `rest.error.message. supported.locales=fr,en,ar`.

6. Select **Customized Upgrade** to proceed with a full installation. During customized installation, all the values from the previous successful installation will be populated automatically. The values can be modified if required. Follow the instructions for upgrading in "Install the client on Documentum CM Server" on page 41.

2.3 Update the application server client configuration

Ensure that you follow the steps listed below before the upgrade:

1. Application server services are stopped before running the installer.
2. Unlike versions before 4.6, do not extract the new .war files before running the installer on the application server.

Upgrading on the application server includes the process of updating the configuration files. This process reads existing configuration files from the web application servers and updates the same files within the new `.war` files. These configuration files include:

- `D2-Config.properties`
- `dfc.properties`
- `logback.xml`
- `settings.properties`
- `D2FS.properties`
- `shiro.ini`

Notes

- If you upgrade the client but not Documentum CM Server, an older version of the `dfc-xx.jar` will be referenced, causing some issues such as search failure in a doclist widget that features custom columns. Ensure that `dfc-xx.jar` is up to date in `<appServer>\webapps\D2\WEB-INF\lib`.
- As of version 4.7, `D2FS-trust.properties` was made obsolete and has been replaced by the keystore.
- The names of configuration files are case-sensitive.

For the *<D2FS.properties>*, *<D2-Config.properties>*, and *<settings.properties>* files, the old configuration values are merged with the values in the new configuration files. The rest of the configuration files are copied and replaced with the new configuration files in the newly created WAR files. All the standard files are backed up with a *<.bak>* extension. On every upgrade, a new backup is generated as: *<<name>.jar.bak.0>*, *<<name>.jar.bak.1>*, and so on.

To update application server configuration files:

1. Keep `D2-Config.war`, `D2.war`, and `D2-Smartview.war` in a separate folder. During the upgrade, you will be asked the location of your existing `.war` files and the location of your new `.war` files so the process can upgrade/merge the configuration files.
2. Clear the cache and the temporary folder.
   a. On Apache Tomcat:

i. Clear the Catalina cache in the folder *<install path of Tomcat>*\work\Catalina\localhost\

ii. Clear the Tomcat temporary folder *<install path of Tomcat>*\temp\

b. On Wildfly, clear the temp folder <install path of the web application server>/standalone/tmp.

3. Extract Documentum-Client-Installer-*<version>*.zip.

   Double-click the **Documentum-Client-Installer** application to launch installer and then click **Next**.

   Note: After installing, ensure you grant the applications "Privileged DFC client" access in your repositories. See section 14.10 "Privileged clients" in *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* for instructions.

4. Select **Update Application Server Documentum Client Configuration Files**. Click **Next**.

5. Specify the location of the old webapps folder *<install path of the web application server>*\webapps where the old configurations files are located. Click **Next**.

6. Specify the location where new D2.war, D2-Smartview.war, and D2-Config.war are located. Click **Install**.

7. Clear the Appserver cache and the browser cache.

8. Approve the new dfc.keystore.

9. Start the Application Server.

2.4 Installation validation

Use this topic to validate the installation by verifying that the installation process correctly extracted and deployed the necessary files and folders. This topic does not include library files installed by plug-ins.

Make sure the .dar files in the dar folder are deployed by checking the dar logs present in the folder: <documentum install folder>\dba\config\<reponame>\

Libraries in the Java Method Server on the Documentum CM Server host

Verify the following files in the *<installation path of Documentum>*\Tomcat *<Version>*\webapps\DmMethods\WEB-INF\lib folder.

- activation.jar
- avalon-framework-*<version>*.jar
- batik-all-*<version>*.jar

- bcmail-jdk<*version*>.jar
- bcprov-jdk<*version*>.jar
- C6-Common-<*version*>.jar
- commons-collections-<*version*>.jar
- commons-codec-<*version*>.jar
- commons-compress-<*version*>.jar
- commons-io-<*version*>.jar
- commons-lang-<*version*>.jar
- commons-math3-<*version*>.jar
- cryptoj.jar
- ctsTransform-<*version*>.jar
- curvesapi-<*version*>.jar
- D2-API-<*version*>.jar
- D2BofServices-<*version*>.jar
- D2CacheManager-<*version*>.jar
- D2-Constants-<*version*>.jar
- D2Core-<*version*>.jar
- D2Methods-<*version*>.jar
- D2-Widget-API-<*version*>.jar
- D2FS-Generated-<*version*>.jar
- D2FS4DCTM-API-<*version*>.jar
- D2-Specifications-API-<*version*>.jar
- D2-Specifications-Install-<*version*>.jar
- D2-Specifications-Plugin-<*version*>.jar
- D2TBOs-<*version*>.jar
- D2-Widget-Install-<*version*>.jar
- D2-Widget-Plugin-<*version*>.jar
- dom4j-<*version*>.jar
- dtdparser<*version*>.jar
- ehcache-core-<*version*>.jar
- fop-core-<*version*>.jar
- fop-events-<*version*>.jar
- fop-hyph.jar

- fop.jar
- fop-util-<*version*>.jar
- janino-<*version*>.jar
- jcl-over-slf4j-<*version*>.jar
- jcifs-krb5-<*version*>.jar
- jul-to-slf4j-<*version*>.jar
- krbutil.jar
- logback-classic-<*version*>.jar
- logback-core-<*version*>.jar
- opencsv-<*version*>.jar
- org.suigeneris.jrcs.diff-<*version*>.jar
- PDF-API.jar
- poi-<*version*>.jar
- poi-excelant-<*version*>.jar
- poi-ooxml-<*version*>.jar
- poi-ooxml-schemas-<*version*>.jar
- poi-scratchpad-<*version*>.jar
- README.txt
- serializer-<*version*>.jar
- slf4j-api-<*version*>.jar
- wfde.jar
- xalan-<*version*>.jar
- xmlbeans-<*version*>.jar
- xmlgraphics-commons-<*version*>.jar
- spring-security-oauth2-client-<*version*>.jar
- spring-security-oauth2-core-<*version*>.jar
- spring-security-core-<*version*>.jar
- reactive-streams-<*version*>.jar
- reactor-core-<*version*>.jar
- httpclient5-<*version*>.jar
- httpcore5-<*version*>.jar
- httpcore5-h2-<*version*>.jar
- httpcore5-reactive-<*version*>.jar

- jackson-databind-<*version*>.jar
- bpm_infra_da-<*version*>.jar
- javax.mail-<*version*>.jar
- xml-apis-ext-<*version*>.jar

Verify the following OpenText Documentum Content Management (CM) Foundation SOAP API files:

- emc-collaboration-services.jar
- emc-collaboration-services-remote.jar
- emc-dfs-rt.jar
- emc-dfs-services.jar
- collaboration.jar
- configservice-api.jar
- configservice-impl.jar
- dfc.jar
- dms-client-api.jar
- xtrim-api.jar
- xtrim-server.jar
- jaxb-api.jar
- jaxb-impl.jar
- jaxws-api.jar
- jaxws-rt.jar
- jsr<*version*>_api.jar
- jsr<*version*>-api.jar
- stax-ex.jar
- aspectjrt.jar
- log4j.jar

If you installed a version of Federal Information Processing Standards (FIPS) older than Foundation SOAP API version 7:

- certjFIPS.jar
- jsafeFIPS.jar

If you installed FIPS Foundation SOAP API 7:

- certj.jar

- cryptoFIPS.jar

If you installed FIPS Foundation SOAP API 7.1:

- jcmFIPS.jar
- certj.jar
- cryptojce.jar
- cryptojcommon.jar

Business Process Management on the Documentum CM Server host

If you installed with Business Process Management, verify the following files in the *<installation path of Documentum>*\Tomcat *<Version>*\webapps\bpm\WEB-INF\lib folder.

- activation.jar
- avalon-framework-<*version*>.jar
- batik-all-<*version*>.jar
- bcmail-jdk<*version*>.jar
- bcprov-jdk<*version*>.jar
- C6-Common.jar
- commons-collections-<*version*>.jar
- commons-io-<*version*>.jar
- commons-lang-<*version*>.jar
- commons-codec-<*version*>.jar
- commons-compress-<*version*>.jar
- commons-math3-<*version*>.jar
- cryptoj.jar
- curvesapi-<*version*>.jar
- D2-API.jar
- D2BofServices-<*version*>.jar
- D2CacheManager-<*version*>.jar
- D2Core-<*version*>.jar
- D2FSDCTM-API.jar
- D2FS-Generated.jar
- D2Methods-<*version*>.jar
- dtdparser<*version*>.jar

- D2TBOs-<*version*>.jar
- ehcache-core-<*version*>.jar
- fop-hyph.jar
- fop.jar
- janino-<*version*>.jar
- javax.mail-<*version*>.jar
- jaxb-impl-<*version*>.jar
- jaxb-xjc.jar
- jcl-over-slf4j-<*version*>.jar
- jul-to-slf4j-<*version*>.jar
- jcifs-krb<*version*>.jar
- krbutil.jar
- logback-classic-<*version*>.jar
- logback-core-<*version*>.jar
- opencsv-<*version*>.jar
- org.suigeneris.jrcs.diff-<*version*>.jar
- PDF-API.jar
- poi<*version*>.jar
- poi-ooxml-<*version*>.jar
- poi-ooxml-schemas-<*version*>.jar
- poi-scratchpad-<*version*>.jar
- README.txt
- slf4j-api-<*version*>.jar
- xmlgraphics-commons-<*version*>.jar

Verify the following Foundation SOAP API files:

- aspectjrt.jar
- collaboration.jar
- configservice-api.jar
- configservice-impl.jar
- cryptoj.jar
- dfc.jar
- dms-client-api.jar
- emc-dfs-rt.jar

- emc-dfs-services.jar
- jaxb-api.jar
- jaxb-impl.jar
- jaxb-xjc.jar
- jaxws-api.jar
- jaxws-rt.jar
- jsr<*version*>_api.jar
- jsr<*version*>-api.jar
- log4j.jar

If you installed a version of Federal Information Processing Standards (FIPS) older than Foundation SOAP API version 7:

- certjFIPS.jar
- jsafeFIPS.jar

If you installed FIPS Foundation SOAP API 7:

- certj.jar
- cryptoFIPS.jar

If you installed FIPS Foundation SOAP API 7.1:

- jcmFIPS.jar
- certj.jar
- cryptojce.jar
- cryptojcommon.jar

WAR files on the web application server host

Navigate to `<APPSERVER_INSTALLATION_PATH>` and verify that it contains `D2.war` (for Classic View), `D2-Smartview.war` for Smart View, and `D2-Config.war`. For example, in Tomcat the `D2.war` should be located in the `webapps` folder.

When the web application server is running, verify that the `.war` files were deployed to the `D2` and `D2-Config` folders. Depending on your web application server, you may need to perform additional steps to deploy the `.war` files.

To resolve a cross-site access issue with Chrome:

In HTTPS-enabled cross-site access scenarios, Chrome version 84 and later expects all cookies to have `SameSite=None` and `Secure` flags. If these flags are not available, the cookies get rejected by the browser. This impacts the loading of the application when it is invoked from other applications. To resolve this issue, perform the following steps:

1. Edit `ESAPI.properties` (`D2.war\WEB-INF\classes`) and update `HttpUtilities.ForceSecureCookies` to `true`. This will make sure the cookies created by the application are set `Secure` and cross-site access is available.
2. For Tomcat, create `context.xml` with the following content and place it under `D2.war\META-INF`:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
  <CookieProcessor sameSiteCookies="None" />
</Context>
```

2.5 Run the Workflow Migration utility

In version 24.4 of client configuration workflow configuration, several updates and internal schema changes were introduced, primarily to enhance compatibility and future functionalities.

Post-installation changes

Once installation is complete and workflow configurations are imported, the new configurations are automatically applied. Typically, no additional actions are needed for basic configurations.

Versioned process templates

If process templates are versioned, it is mandatory to run the Migration utility to ensure that the configuration updates are properly applied.

The command to run the migration utility is:

```
D2workflowConfigMigrationutil.sh -login <username> -password <password> -docbase
<repository> -migrate_assoc_process_version yes
```

Versioned process

When you deploy a process template, it automatically creates a dm process object with version 1.0 by default.

However, if you have created a version of a process template using legacy designers like xCP designer, OpenText Documentum Content Management (CM) Advanced Workflow designer, and Process Builder, it is treated as a versioned process. If you've used such versioning methods, you need to run the Migration utility to ensure the proper configuration is updated.

Note: Ensure that the Migration utility is run when upgrading to 24.4.

Export and import configuration recommendations

After using the Migration utility to migrate the workflow configurations to version 24.4, you do not need to run the Migration utility again. Instead, it is advisable to export the existing configurations and to utilize this package for importing into

other environments running version 24.4 or later. This approach will eliminate the need to rerun the Migration utility in future versions.

Chapter 3

Install the client on Documentum CM Server

The following instructions contain the steps for installing the client on Documentum CM Server.

Note: If you are using Red Hat® WildFly or Apache® Tomcat™ to deploy, you must update the java policy file.

Update the JRE policy by editing or creating the `java.policy` file (`/lib/security/java.policy`) with the following permission:

```
grant {
  permission com.documentum.fc.client.impl.bof.security.RolePermission "*", "propagate";
};
```

`RolePermission` is propagated to the `permissions.xml` file by the installer as follows::

```
<?xml version="1.0" encoding="UTF-8"?>
<permissions xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://
xmlns.jcp.org/xml/ns/javaee/ permissions_7.xsd" version="7">
    <permission>
        <class-name>com.documentum.fc.client.impl.bof.security.RolePermission</class-
name>
        <name>*</name>
        <actions>propagate</actions>
    </permission>
</permissions>
```

1. Extract `Documentum-Client-Installer-<version>.zip`.

   The ZIP file contains the following:

   - `Dars.zip`
   - `Documentum-Client—Installer.bin`
   - `Documentum-Client-Installer.exe`
   - `LanguagePacks.zip`
   - `libraries.zip`
   - `MANIFEST.MF`
   - `Plugins.zip`
   - `SilentInstall.properties`
   - `Webapps.zip`

   Double-click the **Documentum-Client-Installer** application to launch installer and then click **Next**.

Notes

- If you do not have administrator rights, select **Run installer as administrator**.
- Run any permissions in the default temp directory of the host.

- *For Microsoft Windows:*

  The environment installer uses the `java.io.tmpdir` Java temporary directory for Java Virtual Machine (JVM) as its temporary directory:

  `C:\Users\<username>\Local Settings\Temp\DCTMCM_<version>`

  <*username*> is the user name of the account, and <*version*> is the version number.

- *For Linux Environment:*

  Open an xterm and run the installer by typing:

  `./Documentum-Client-Installer.bin`

  The environment installer uses the `java.io.tmpdir` Java temporary directory for Java Virtual Machine (JVM) as its temporary directory:

  `/tmp/DCTMCM_<version>`

2. Select **I accept the terms of the license agreement** to continue with the installation.
3. Select **Install Documentum Client** to start the installation on Documentum CM Server, and then click **Install**.
4. On the **Select installation components** panel, select **Classic View**, **Client Config**, **Smart View**, **Admin Console**, **Client API for Java Method Server (JMS)**, **Client API for Business Process Manager (BPM)**, and **DAR**. Click **Next**.

   Note: To install APIs for BPM, you must have the Documentum Process Engine installed on your machine.

5. On selection of **Smartview pack**, the **Smartview Configuration Panel** page opens. On this page, specify the crypto salt value for client token encryption and decryption. These values are updated in the `rest-api-runtime.properties` file. Clear the check box to remove the Swagger Documentation from `D2-Smartview/rest`. Click **Next**.
6. On the **WebApp and DAR extraction folder** panel, for WAR file(s) installation, the destination path will be auto-populated or the user can select a folder to which the installer extracts `D2.war`, `D2-Smartview.war`, `admin-console.war`, and `D2-Config.war`. The selected folder must not already contain these WAR files.

   For DAR file(s) installation, the destination path will be auto-populated or the user can select the path to which the installer extracts the DAR files. Click **Next**.

This step generates the brava_formats.dar, D2-DAR.dar, D2Widget-DAR.dar, D2Widget-DAR_DB2.dar, OTIVPublication.dar, DCTM_Reports_Base.dar, DCTM_Reports_GBL.dar, and DCTM_Reports_Templates_for_D2.dar and admin-console-dar.dar files in the selected folder.

Note: The installer automatically deploys the DAR files as part of the installation process. Contrary to previous versions, it is now necessary to deploy D2-DAR.dar to the global registry repository. This will also cause themes to be properly and consistently displayed across all repositories.

7. On the plug-ins installation panel, select plug-ins by checking the corresponding check boxes to install new plug-ins or update existing ones. Click **Next**.

   Note: With the new install anywhere framework, plug-ins are bundled with the installer. You no longer need to browse plug-ins in a list to copy the plug-in folder separately.

   You can select **Core Signature**, **DocuSign**, **PDF Configuration**, **Transfer Configuration**, **Recycle Bin**, **Retention Policy Services (RPS)**, **InfoArchive**, **Extended ECM (xECM)**, and **Reports** plug-ins. The installer automatically runs the added plug-in installer files, activates the plug-ins, and automatically deploys DAR files. For example, if you select **PDF Configuration**, you do not need to:

   - deploy the output C2-API.jar and C2-Plugin.jar files,
   - configure D2-Config.properties, or
   - deploy C2-DAR.dar.

   If you do not add plug-ins using the installer, you can manually run the plug-in installer .bin for Linux and .exe for Windows, and the plug-in installer automatically copies the JAR files to the respective folders.

   Notes

   - On selecting the **Reports** plug-in, the plug-in JARs (DCTM-Reports-Plugin and DTR-API) are copied, and the respective DARs (DCTM_Reports_Base.dar, DCTM_Reports_GBL.dar, and DCTM_Reports_Template_for_D2.dar) are deployed.
   - For more information on installing digital signature plug-ins, see "Digital signature components" on page 169.

8. Fill out the **Client Library Extraction Foders** panel, as described in the following table, and click **Next**.

Field	Path
Documentum Client API file destination for Java Method Server	For Documentum CM Server version 21.2 and later, use *<install path of Documentum>**<Java method server>*\webapps\DmMethods\WEB-INF
Documentum Client API file destination for Business Process Manager	For Documentum CM Server version 21.2 and later, use *<install path of Documentum>* *<Java method server>*\webapps\bpm\WEB-INF

9. In the **Install Owner Credentials** panel, enter the Documentum CM Server install owner's name and password. Click **Next**.

   Note: The install owner account must remain active at all times to avoid Authentication Failed errors when performing certain administrative functions in client configuration, such as creating and saving dictionaries.

10. Do the following:

    a. In the **Repository Configuration** panel, select the repositories for which you want to install the client. The client will be deployed on each of the repositories selected here.

    b. Select **Yes** to prevent repeating attributes from being returned as individual rows in lists, such as advanced searches, property pages, and repository browser widgets. Click **Next**.

       Note: Ensure you restart the repositories and docBrokers after installing on Documentum CM Server.

11. Select **Yes** to force the client to apply autolink rules before applying security rules to the content. Select **No** to force the client to apply security rules to the content before applying autolink rules. Click **Next**.

12. When you select the **Classic View**, **Client Config**, or **Smart View** pack, the **Client Language Pack** panel appears.

    Select the required localization by selecting the appropriate check boxes to install the corresponding locales. Click **Install**.

    Note: Install the locale to the repository. The *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* contains future instructions on populating and publishing localized data dictionaries to the repository.

    You can select **Arabic**, **German**, **Spanish**, **French**, **Italian**, **Japanese**, **Korean**, **Dutch**, **Portuguese**, **Swedish**, and **Chinese** locales. The installer automatically copies the locale JARs to the respective WARs.

    For example, if you select **Arabic**, you do not need to:

- copy the locale JARs (`C2-LanguagePack_ar.jar`, `D2-Bin-languagePack_ar.jar`, `D2FS4DCTM-LanguagePack_ar.jar`, `D2-LanguagePack_ar.jar`, and help folder) to `tomcat\webapps\D2\WEB-INF` or `tomcat\webapps\D2\help` or
- copy the locale JARs (`D2FS4DCTM-LanguagePack_ar.jar` and UI folder) to `tomcat\webapps\D2-Smartview\WEB-INF` or `tomcat\webapps\D2-Smartview\ui`.

If **Client Config** and **French** are selected, you do not need to:

- copy the locale JARs (`C2-LanguagePack_fr.jar`, `D2-Bin-languagePack_fr.jar`, `D2FS4DCTM-LanguagePack_fr.jar`, `D2-LanguagePack_fr.jar`, and help folder) to `tomcat\webapps\D2\WEB-INF` or `tomcat\webapps\D2\help`,
- copy the locale JARs (`C2-LanguagePack_fr.jar`, `D2-Bin-languagePack_fr.jar`, `D2FS4DCTM-LanguagePack_fr.jar`, `D2-LanguagePack_fr.jar`, `D2-Config-LanguagePack_fr.jar`, `D2-InfoArchive-LanguagePack_fr.jar`, `D2-RPS-LanguagePack_fr.jar`, `D2-Specifications-LanguagePack_fr .jar`, `D2-xECM-LanguagePack_fr.jar`, `O2-LanguagePack_fr.jar`, and help folder) to `tomcat\webapps\D2-Config\WEB-INF` or `tomcat\webapps\D2-Config\help`, or
- copy the locale JARs (`D2FS4DCTM-LanguagePack_fr.jar`, `D2-InfoArchive-LanguagePack_fr.jar`, `D2-xECM-LanguagePack_fr.jar`, and UI folder) to `tomcat\webapps\D2-Smartview\WEB-INF` or `tomcat\webapps\D2-Smartview\ui`.

Note: Ensure that the following entry in the `rest-api-runtime.properties` file in `D2-Smartview\WEB-INF\classes` displays your intended locales, in short form, separated by commas (default is en). For example: `rest.error.message. supported.locales=fr,en,ar`.

13. When the installation is finished, click **Next** and then click **Done**.

Chapter 4

Silent install and upgrade

The client can be installed and upgraded using a silent install method. To do so, you must run the silent install property file `SilentInstall.properties` from `Documentum-Client-Installer-<version>.zip` through the following command:

<location of Documentum-Client-Installer application> `-i silent -f` *<location of SilentInstall.properties>*

4.1 Silent install

The silent install script walks you through the installation and configuration process. You are prompted to provide the details of your installation.

If you want to use the silent install process to upgrade, see "Silent upgrade" on page 48.

The following are the fields that you need to complete when running the silent install script:

- **License_agreement:** If it is true, indicates acceptance of the license terms, allowing the installer to continue with the installation process.
- **Installer_pack:** Specifies the installer files and should be true always.
- **d2_pack:** If it is true, `D2.war` will be generated.
- **d2config_pack:** If it is true, `D2-Config.war` will be generated.
- **d2sv_pack:** If it is true, `D2-Smartview.war` will be generated.
- **adminconsole_pack:** If it is true, `admin-console.war` will be generated.
- **d2api_pack:** If it is true, the JARs will be copied to the Java Method Server.
- **d2apibpm_pack:** If it is true, the JARs will be copied to Business Process Manager.
- **docappdar_pack:** If it is true, Dars will be deployed.
- **webappsDir:** Specifies the folder location where the WAR files will be copied.
- **pluginInstaller:** Specifies the comma-separated file names of the plug-ins that will be installed.
- **bpmDir:** The bpm directory.
- **jmsDir:** The JMS directory.

Note: In the properties file, use double backslashes for Windows folder paths (e.g., `C:\\path\\to\\webapps`) and forward slashes for Linux folder paths (e.g., `/path/to/webapps`).

- **Selected_LanguagePack:** Specifies the comma-separated names of the locales that will be installed.
- **installationDir:** Specifies the folder location where the DAR files will be copied.
- **DTR-Reports-Plugin:** If it is true, the reports plug-in JARs will be copied.
- **COMMON.USER_ACCOUNT:** Specifies the username.
- **install.owner.password:** Specifies the password of the install owner.
- **SERVER.REPOSITORIES.NAMES:** Comma-separated repository names where you want to deploy.
- **CRYPTOSALTVALUE:** Specifies the crypto salt value for client token encryption and decryption.
- **SWAGGER_DOCUMENTATION:** If it is true, Swagger documentation will be included in `D2-Smartview.war`.

4.2 Silent upgrade

To use the silent install script to upgrade, you must make some changes to the `.properties` file.

Change values as follows:

```
InstallD2=false
UpgradeD2=true
CustomizedUpgrade=true
```

Chapter 5

Licensing OpenText Documentum CM

After installing or upgrading OpenText Documentum CM, you must apply licenses to allow users to access its various components. OpenText Documentum CM uses OpenText™ Directory Services (OTDS) to apply licenses for all the OpenText Documentum CM components.

How to procure the license file and configure OTDS and licenses depends on whether you installed OpenText Documentum CM for the first time or if you upgraded to a new version. See the relevant guide, based on your scenario:

- **Installation**: Section 2.7.1 “Licensing OpenText Documentum CM” in *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)*
- **Upgrade**: Section 4.6.1 “Licensing OpenText Documentum CM” in *OpenText Documentum Content Management - On-Premises Upgrade and Migration Guide (EDCCS-UMD)*

Chapter 6

Deploy DAR files manually

The core installer automatically deploys the following DAR files. If needed, these DAR files can be deployed manually as described in the following table.

DAR	Description
D2-DAR.dar	Deploy to install the core data model to a repository. Contrary to previous versions, it is now necessary to deploy D2-DAR.dar to the global registry repository. This will also cause themes to be properly and consistently displayed across all repositories.
D2Widget-DAR.dar	Deploy to install the client.
Plug-in DAR files, such as C2-DAR.dar.	Deploy to use the respective plug-ins. PDF Configuration, Transfer Configuration, Recycle Bin, and D2-RPS dar files are deployed for the respective plug-ins if indicated during the installation process.

The D2-DAR.dar and D2Widget-DAR.dar files were extracted to the folder you specified during the install process, which is by default *<install path>*/dars.

Note: The installer does not install the Collaboration Services DAR file because the CS installer installs the Collaboration_Services.dar file.

1. Make sure Documentum CM Server services are running.
2. For each DAR file, run the DAR installer shipped with Documentum Composer, dardeployer.exe, and fill out the form as described in the following table:

Field	Description
DAR	Locate and select the DAR file.
Docbroker Details	Select the target Docbroker and port. Click **Connect**.

Field	Description
Repository Details	Select the repository with Documentum CM Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the `nodmadmin.installparam.xml` file if the Documentum CM Server installation owner is not named dmadmin, as described in step 3.

3. If the Documentum CM Server installation owner is not dmadmin:
   a. Create a file in a text editor and save it as `nodmadmin.installparam.xml`
   b. Add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
  <installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:installparam="installparam">
    <parameter key="dmadmin" value="<administrator account name>"/>
  </installparam:InputFile>
```

   c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam.xml` file you created.
4. Click **Install**.
5. Click **Recent DAR install log files** to review log files.
6. Return to the instructions: "Instructions for installing the client" on page 14.

Chapter 7

Install the client on the web application server

Limitations of core installer

By design, the core installer serves dual purposes. The core functionality, which is only available when the install utility is run on a Documentum CM Server machine, is to install by deploying artifacts to Documentum CM Server, generate `D2.war` application files and deploy related Archives (`D2-DAR.dar` and `D2-Widget-DAR.dar` files and optionally other plug-in DAR files) into one or more repositories that project to Documentum CM Server.

7.1 Install on Apache Tomcat

Note: All Tomcat deployments should have the following set in the `server.xml` file:

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b
%D" />

<Valve className="org.apache.catalina.valves.StuckThreadDetectionValve" threshold="30" />
```

Ensure the first valve has `%D` at the end.

Ensure the second valve is included and uncommented.

To support Java 17, see "Prepare Java runtime" on page 62.

1. Copy `D2-Config.war`, `D2-Smartview.war`, and `D2.war` to the *<install path of the web application server>*`/webapps` folder.
2. If during the install wizard you did not place the configuration files in the default locations:
   - Copy the configuration files to the *<install path of Tomcat>*`/webapp/D2-Config/WEB-INF/classes` folder for manual deployment, or
   - Update the references to where the configuration files are located.

   To update the references:

   a. Navigate to *<install path of Tomcat>*`/work/Catalina/conf/` and open the `catalina.properties` file.

   b. Find the line `common.loader=`

   c. To use a common `dfc.properties` file D2-Config, append the location of `dfc.properties`

For example, `common.loader=<existing paths>,<install path of Documentum>/Config`

3. Add or increase the following Java options in your application server environment to instruct the JVM to configure Metaspace:

   `-XX:MetaspaceSize=<YYY>m` (for example 256m): sets the initial size of the permanent generation memory space upon startup of Tomcat.

   `-XX:MaxMetaspaceSize=<YYY>m` (for example 256m): sets the maximum amount of permanent generation memory space that can be allocated.

   Set `MetaspaceSize` to the same value as `MaxMetaspaceSize` to allocate the maximum amount of permanent generation memory from startup to help reduce the occurrence of full garbage collection.

   You can also configure the clearing of classes by using the following command:

   - CMSPermGenSweepingEnabled: `-XX:+CMSPermGenSweepingEnabled`
   - CMSClassUnloadingEnabled: `-XX:+CMSClassUnloadingEnabled`

   https://docs.oracle.com/en/java/javase/{javaversion}/gctuning/introduction-garbage-collection-tuning.html contains further information about JVM garbage collection settings.

4. Add the following Java option in your application server environment to instruct the JVM to use CMS GC instead of G1 GC. Note that this change is applicable to both Classic View and Smart View:

   `-XX:+UseParallelGC`

5. In order to change the path of the `trace.log` file or modify the logging properties after installation, copy `log4j.properties` to the `Tomcat` folder. The `trace.log` file has the potential to grow to unmanageable size and cause performance issues in the production environment unless you perform mitigation steps.

6. Return to the instructions: "Instructions for installing the client" on page 14.

7.2 Install on Wildfly or Red Hat JBoss Server

Notes

- If you are using JBoss or Wildfly to deploy you will have to update the java policy or `permissions.xml` file first.

  If you are using JBoss or Wildfly and Documentum CM Server 23.4 and above on a Windows machine, add the `permissions.xml` in `C:\Documentum\{Tomcat version}\webapps\DmMethods\META-INF` and `C:\Documentum\{Tomcat version}\webapps\bpm\META-INF` on Documentum CM Server and the META-INF folder on the application server to give propagate action for RolePermission:

```
<?xml version="1.0" encoding="UTF-8"?>
<permissions xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://
```

```
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/
javaee http://xmlns.jcp.org/xml/ns/javaee/ permissions_7.xsd" version="7">
    <permission>
        <class-name>com.documentum.fc.client.impl.bof.security.RolePermission</
class-name>
        <name>*</name>
        <actions>propagate</actions>
    </permission>
</permissions>
```

Make sure the `<permissions>` element is all on a single line otherwise the `permissions.xml` file will not be properly read by the server.

- To support Java 17, see "Prepare Java runtime" on page 62.
- The web app needs to be registered as a privileged client.
- To improve the response time of the Smart View client, make the following changes in Java:
  1. Open the `$JAVA_HOME/jre/lib/security/java.security` file in a text editor.
  2. Edit the line:

     `securerandom.source=file:/dev/urandom`

     To:

     `securerandom.source=file:/dev/./urandom`
  3. Add the following lines in the `jboss-deployment-structure.xml` present in META-INF of the Smart View `.war` file before deployment:

     ```
     <exclude-subsystems>
       <subsystem name="jaxrs"/>
       <subsystem name="webservices"/>
     </exclude-subsystems>
     ```
  4. Restart the JBoss server.
- At present, Wildfly is using an integer range validator to validate the `max-post-size` parameter. There is a requirement to support uploading of larger file sizes (this limitation is due to the integer range validator), therefore you need to change the value of `max-post-size` in the `standalone.xml` file where you can configure the value according to your needs.

Perform the following actions to install:

1. Stop the JBoss or Wildfly service.
2. Edit the <*standalone.xml*> file under `<$WILDFLY_HOME>/standalone/configuration` or `<jboss-hone>/standalone/configuration`and replace `127.0.0.1` with the host IP address in the `<wsdl-host>` section and in the `<interfaces>` section as follows:

   ```
   <wsdl-host>${jboss.bind.address:<jboss-host-ip>}</wsdl-host>
   <interfaces>
      <interface name="management">
         <inet-address value="${jboss.bind.address.management:<jboss-host-ip>}"/>
      </interface>
      <interface name="public">
         <inet-address value="${jboss.bind.address:<jboss-host-ip>}"/>
   ```

```
    </interface>
    <interface name="unsecure">
       <inet-address value="${jboss.bind.address.unsecure:<jboss-host-ip>}"/>
    </interface>
 </interfaces>
```

a. Make sure `D2/WEB-INF/`, `D2-Config/WEB-INF` directory has `jboss-deployment-structure.xml` file. The contents of the file are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure xmlns="urn:jboss:deployment-structure:1.2">
  <deployment>
    <exclude-subsystems>
      <subsystem name="jaxrs"/>
      <subsystem name="webservices"/>
    </exclude-subsystems>
    <exclusions>
      <module name="org.slf4j"/>
      <module name="org.apache.log4j"/>
    </exclusions>
    <resources>
      <resource-root path="WEB-INF/lib/cryptoj.jar" use-physical-code-
source="true"/>
      <resource-root path="WEB-INF/lib/cryptojce.jar" use-physical-code-
source="true"/>
      <resource-root path="WEB-INF/lib/cryptojcommon.jar" use-physical-code-
source="true"/>
      <resource-root path="WEB-INF/lib/jcmFIPS.jar" use-physical-code-
source="true"/>
    </resources>
  </deployment>
</jboss-deployment-structure>
```

b. For Smart View deployments on Wildfly, copy `anonymous-service-handler-chain.xml` from `wildflyXX.X.X\server\DctmServer_BOCS\deployments\bocs.ear\lib\configs.jar` and add it to the following location:

`/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/core/anonymous-service-handler-chain.xml`

If you do not have access to `wildflyXX.X.X\server\DctmServer_BOCS\deployments\bocs.ear\lib\configs.jar`, you can copy the file contents from here:

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-
class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler</handler-
class>
    </handler>
  </handler-chain>
</handler-chains>
```

c. For Smart View deployments on Wildfly, copy `authorized-service-handler-chain.xml` from `wildflyXX.X.X\server\DctmServer_BOCS\deployments\bocs.ear\lib\configs.jar` and add it to the following locations:

```
/WEB-INF/lib/D2FS-Generated-21.4.0.jar/com/emc/d2fs/dctm/api/services/ws/
authorized-service-handler-chain.xml
/WEB-INF/lib/emc-collaboration-services.jar/com/emc/documentum/fs/services/
collaboration/authorized-service-handler-chain.xml
/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/core/
```

```
authorized-service-handler-chain.xml
/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/core/vdm/
authorized-service-handler-chain.xml
/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/core/acl/
authorized-service-handler-chain.xml
/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/core/
lifecycle/authorized-service-handler-chain.xml
```

If you do not have access to `wildflyXX.X.X\server\DctmServer_BOCS\deployments\bocs.ear\lib\configs.jar`, you can copy the file contents from here:

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>Authorization</handler-name>
      <handler-
class>com.emc.documentum.fs.rt.impl.handler.AuthorizationHandler</handler-
class>
    </handler>
<!-- handler> <handler-name>Privileged DFC</handler-name> <handler-
class>com.emc.documentum.fs.rt.handlers.PrivilegedDfcHandler</handler-class> </
handler -->
<!-- Any handler using ContextFactory, like KerberosTokenServerHandler or
AuthorizationHandler must be inserted above this comment -->
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-
class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler</handler-
class>
    </handler>
<!-- Any handler modifying DFS SOAP headers must come below this comment -->
  </handler-chain>
</handler-chains>
```

3. Add the following exclusion to the `jboss-deployment-structure.xml` file under both “META-INF” and “WEB-INF”:

```
<exclusions>
    <module name="org.slf4j"/>
    <module name="org.apache.log4j"/>
    <module name="org.apache.logging.log4j.api"/>
    <module name="org.apache.logging.log4j.core"/>
</exclusions>
```

4. Make sure the client and client configuration applications are in exploded format with the folder names as `D2.war` and `D2-Config.war`. Place the exploded folder in the standalone/deployments directory of Wildfly or JBoss.

5. For Wildfly or JBoss to pick up and deploy the exploded directories, create a file called `D2.war.dodeploy` for the client and a file called `D2-Config.war.dodeploy` for client configuration in the same standalone/deployments directory.

   Notes

   - The `dodeploy` files are used to instruct Wildfly or JBoss to deploy the exploded directory.
   - Make sure the exploded directory contains all necessary files and is structured correctly.
   - Monitor the `server.log` file to confirm the deployment.

6. Restart the Wildfly or JBoss service.

Note: When deploying WAR files in exploded format, Wildfly or JBoss automatically detects and deploys them in the Admin Console. No further manual steps are needed, and the application is deployed directly from the standalone/deployments directory.

7. Return to "Instructions for installing the client" on page 14.

7.3 Install on IBM WebSphere Liberty

1. Copy `D2-Config.war`, `D2-Smartview.war`, and `D2.war` to the *<install path of the web application server>*`\dropins` folder.
2. In the `server.xml` file, add the following setting:

   ```
   <classloading useJarUrls="true"/>
   ```

3. Before deploying `D2-Smartview` in the server, make changes to the existing jars as shown below:

   - `anonymous-service-handler-chain.xml` needs to be present in the following location:

     ```
     D2-Smartview/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/
     core/anonymous-service-handler-chain.xml.
     ```

     The content for `anonymous-service-handler-chain.xml` is the following:

     ```
     <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
       <handler-chain>
         <handler>
           <handler-name>Context Local Registry</handler-name>
           <handler-
     class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler</handler-
     class>
         </handler>
       </handler-chain>
     </handler-chains>
     ```

   - `authorized-service-handler-chain.xml` needs to be present in the following location:

     ```
     D2-Smartview/WEB-INF/lib/emc-collaboration-services.jar/com/emc/documentum/fs/
     services/collaboration/authorized-service-handler-chain.xml.
     D2-Smartview/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/
     core/authorized-service-handler-chain.xml
     D2-Smartview/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/
     core/lifecycle/authorized-service-handler-chain.xml
     D2-Smartview/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/
     core/vdm/authorized-service-handler-chain.xml
     D2-Smartview/WEB-INF/lib/D2FS-Generated.jar/com/emc/d2fs/dctm/api/services/ws/
     authorized-service-handler-chain.xml
     D2-Smartview/WEB-INF/lib/emc-dfs-services.jar/com/emc/documentum/fs/services/
     core/acl/authorized-service-handler-chain.xml
     ```

     The content for `authorized-service-handler-chain.xml` is the following:

     ```
     <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
       <handler-chain>
         <handler>
           <handler-name>Authorization</handler-name>
           <handler-
     class>com.emc.documentum.fs.rt.impl.handler.AuthorizationHandler</handler-
     class>
     ```

```
    </handler>
<!-- handler> <handler-name>Privileged DFC</handler-name>
<handlerclass>com.emc.documentum.fs.rt.handlers.PrivilegedDfcHandler</handler-
class></handler -->
<!-- Any handler using ContextFactory, like KerberosTokenServerHandler or
AuthorizationHandler must be inserted above this comment -->
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-
class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler</handler-
class>
    </handler>
<!-- Any handler modifying DFS SOAP headers must come below this comment -->
  </handler-chain>
</handler-chains>
```

4. Start the IBM WebSphere Liberty server by running the following command in the console (console path IBM\WebSphere\Liberty\bin):

   `server.bat start libertysvr1`

Note: If you install the Japanese language pack and cannot create a creation profile, then add the following to the `server.xml` file and then restart the webapp:

```
<webContainer setContentLengthOnClose="false" />
```

7.3.1 Install on IBM WebSphere Liberty with Java

1. Download and install IBM WebSphere Application server Liberty from IBM Installation Manager.
2. Install Java and set JAVA_HOME in Environment Variables.
3. Create a server and, in `server.xml` (`IBM\WebSphere\Liberty\usr\servers\server_name`), add the below feature settings:

```
<feature>jakartaee-10.0</feature>
<httpEndpoint id="defaultHttpEndpoint"
              host="your_hostname"
              httpPort="9080"
              httpsPort="9443" />
```

4. To support Java 17, see "Prepare Java runtime" on page 62.
5. To enable SSL, create a certificate and add the certificate in the path `IBM\WebSphere\Liberty\usr\servers\server_name\resources\security` and additionally add the following tags in the `server.xml` file:

```
<feature>transportSecurity-1.0</feature>
<keyStore id="defaultKeyStore"
          location="certificate_name.jks"
          type="JKS" password="your_password" />
```

6. Set the hostnames in the hosts file.
7. Start the server.

   IBM WebSphere Liberty will be started with Java.

Note: You can also download IBM WebSphere Application server Liberty Jakarta EE 10 from the official site at https://www.ibm.com/support/pages/websphere-liberty-developers.

7.4 Install the OpenText Documentum Content Management (CM) Client REST API

1. Download the Client REST API .war file from the download site.
2. Copy the .war file to the web server and extract it to a folder using the command:

   `jar -xvf <filename>.war`

   Extracted contents are the `META-INF`, `public`, and `WEB-INF` folders.
3. Configure `\\WEB-INF\classes\dfc.properties`, which contains information about the repository, machine, username and password.
   - If you are not installing Client REST API on the same server as Documentum CM Server, configure `\\WEB-INF\classes\dfc.properties` to `CS`.
   - If you are installing Client REST API on the same server as Documentum CM Server, add the following to `\\WEB-INF\classes\dfc.properties: #include C:\documentum\config\dfc.properties.`
4. In order to enable the Client REST API server to load various caches when it starts up, open its corresponding `D2FS.properties` file and update the `LoadOnStartup` parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example: `LoadOnStartup=repo1,repo2`. Note that the corresponding username and password properties for each listed repository need to be set in the global registry keystore. For example,

   ```
   LoadOnStartup.repo1.username=dmadmin1
       LoadOnStartup.repo1.password=password1
       LoadOnStartup.repo2.username=dmadmin2
       LoadOnStartup.repo2.password=password2
   ```

   Or, if the listed repositories all have common admin credentials,

   ```
   LoadOnStartup.*.username=dmadmin
       LoadOnStartup.*.password=password
   ```

   Note: Workflow reporting features might not operate as expected for completed or aborted Workflows if `LoadOnStartup` is not enabled.
5. Navigate to `WEB-INF/classes/rest-api-runtime.properties.template` and note the settings for enabling cross-origin resource sharing and CSRF token settings. Update the same settings to match in the `WEB-INF/classes/rest-api-runtime.properties` file.
6. Once CS is installed in secure mode, copy security certificates (servercrt and brokercrt) files from `C:\Documentum\dba\secure\`to Foundation Java API client machine.

7. Run the following commands to import CS and Docbroker certificates in java used by Foundation Java API clients:

```
cmd> keytool -import -trustcacerts -file <path_to_brokercrt.der_location> -keystore
<JAVA_HOME>\lib\security\cacerts -alias broker

keytool -import -trustcacerts -file <path_to_servercrt.der_location> -keystore
<JAVA_HOME>]\lib\security\cacerts -alias server

password - changeit
```

Note: If Java home has space like program files, add path in quotes:

```
keytool -import -trustcacerts -file C:\servercrt.der -keystore "C:\Program Files
\Java\{Java version}\lib\security\cacerts" -alias server
```

8. Verify the Client REST API installation by accessing the URL in this format: `http://ServerIPAddress:PortNumber/ExtractedFoldername/services`. Ensure the Client REST API URL is publicly accessible through one of the deployment options.

Note: For on-prem deployed Client REST API, the Swagger documentation, if not required, can be removed by deleting the `openapi` folder (which holds the Swagger documentation) inside the `public` folder after deployment.

7.5 Configure the Client REST API extension framework

Deploy the Client REST API .war on the application server and perform the following post installation steps:

1. Update `dfc.properties` in `\WEB-INF\classes`.
2. Update OpenText Documentum Content Management (CM) Accelerated Content Services or OpenText Documentum Content Management (CM) Branch Office Caching Services in `\WEB-INF\classes\D2FS.properties` for Accelerated Content Services and Branch Office Caching Services environments.
3. Set the client URL in `WEB-INF\classes\settings.properties`.
4. Copy C2 jars (`C2-API.jar` and `C2-Plugin.jar`) to `\WEB-INF\lib` to configure the PDF Configuration plug-in.

7.6 Prepare Java runtime

Make the following changes. Instructions differ depending on your operating system.

Tomcat

For Windows machines, add the following lines to `tomcat\bin\catalina.bat` under "rem Configure JAVA OPTION specific start-up parameters":

```
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 "JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8"
set "JAVA_OPTS=%JAVA_OPTS% -Djava.locale.providers=COMPAT,SPI"
```

If you use a Windows service to start Tomcat, update the `tomcat\bin\tomcat10w.exe` file. Open `tomcat10w` and under **Java9 options** on the **Java** tab, use the following to copy and paste into it:

```
--add-opens=java.base/java.lang=ALL-UNNAMED
--add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
--add-exports=java.base/sun.security.provider=ALL-UNNAMED
--add-exports=java.base/sun.security.pkcs=ALL-UNNAMED
--add-exports=java.base/sun.security.x509=ALL-UNNAMED
--add-exports=java.base/sun.security.util=ALL-UNNAMED
--add-exports=java.base/sun.security.tools.keytool=ALL-UNNAMED
--add-opens=java.xml.crypto/com.sun.org.apache.xml.internal.security=ALL-UNNAMED
-Dfile.encoding=UTF-8
```

For Linux machines, add the following lines to `catalina.sh` under "Add the security start-up parameters required by Tomcat":

```
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.net=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.lang.ref=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-opens=java.naming/com.sun.jndi.toolkit.url=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports java.base/sun.security.provider=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports java.base/sun.security.pkcs=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports java.base/sun.security.x509=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports java.base/sun.security.util=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports java.base/sun.security.tools.keytool=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-opens=java.xml.crypto/
com.sun.org.apache.xml.internal.security=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8"
```

Use the following to copy and paste into the `catalina.sh` control script:

```
--add-opens=java.base/java.net=ALL-UNNAMED
--add-opens=java.base/java.lang.ref=ALL-UNNAMED
--add-opens=java.naming/com.sun.jndi.toolkit.url=ALL-UNNAMED
--add-exports java.base/sun.security.provider=ALL-UNNAMED
--add-exports java.base/sun.security.pkcs=ALL-UNNAMED
```

```
--add-exports java.base/sun.security.x509=ALL-UNNAMED
--add-exports java.base/sun.security.util=ALL-UNNAMED
--add-exports java.base/sun.security.tools.keytool=ALL-UNNAMED
--add-opens=java.xml.crypto/com.sun.org.apache.xml.internal.security=ALL-UNNAMED
```

IBM Liberty

Add the following lines to `server.bat`:

```
set JAVA_ARGS="-Dcom.ibm.jsse2.overrideDefaultTLS=true"
set "JAVA_OPTS=-Dprogram.name=%PROGNAME% %JAVA_OPTS%
set "JDK_JAVA_OPTIONS=-Dprogram.name=%PROGNAME% %JDK_JAVA_OPTIONS%"
set "JAVA_OPTS=%JAVA_OPTS% --add-opens=java.base/java.lang=ALL-UNNAMED"
set "JAVA_OPTS=%JAVA_OPTS% --add-opens=java.base/java.io=ALL-UNNAMED"
set "JAVA_OPTS=%JAVA_OPTS% --add-opens=java.base/java.util=ALL-UNNAMED"
set "JAVA_OPTS=%JAVA_OPTS% --add-opens=java.base/java.util.concurrent=ALL-UNNAMED"
set "JAVA_OPTS=%JAVA_OPTS% --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED"
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.base/java.lang=ALL-UNNAMED"
set "JDK_JAVA_OPTIONS=%JDK_JAVA_OPTIONS% -Dfile.encoding=UTF-8"
```

Wildfly Server

Add JAVA_TOOL_OPTIONS as environment variable:

```
--add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-
UNNAMED --add-exports=java.base/sun.security.provider=ALL-UNNAMED --add-
exports=java.base/sun.security.pkcs=ALL-UNNAMED --add-exports=java.base/
sun.security.x509=ALL-UNNAMED --add-exports=java.base/sun.security.util=ALL-UNNAMED --
add-exports=java.base/sun.security.tools.keytool=ALL-UNNAMED
-Dfile.encoding=UTF-8
```

Oracle Weblogic

Not supported with Java 17.

Note: For Java 21, include the following JVM option in Java runtime:

```
-Djava.locale.providers=COMPAT,SPI
```

Chapter 8

Configure the client

8.1 Configure client configuration

1. Navigate to the location of your client configuration configuration files.

   The default location is *<install path to web application server>*/webapps/D2-Config/WEB-INF/classes

2. Configure dfc.properties:

   a. If you want to use a shared set of configurations, configure dfc.properties to refer to an existing dfc.properties

      By default, dfc.properties contains a reference to the client dfc.properties file. All settings found in the referenced dfc.properties apply to client configuration. Do not remove the # as the full command is #include, and the line is not being commented out.

      #include *<install path to Documentum>*/config/dfc.properties

   b. If you want to create application-specific settings that override the shared dfc.properties, append the settings to the dfc.properties found in *<install path to web application server>*/webapps/D2-Config/WEB-INF/classes.

      See *<install path to Documentum>*/config/dfcfull.properties for possible settings.

   c. Ensure that the dfc.properties file being used or referred to addresses the correct docbroker and port:

      dfc.docbroker.host=*<IP address of the Fully Qualified Domain Name of the docbroker host>*

      dfc.docbroker.port=*<port>*

   d. If your Documentum CM Server installation uses non-anonymous certificates, copy dfc.keystore from the $dm_home\dba\secure folder on the Documentum CM Server machine. The *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)* contains more information on the Documentum CM Server keystore.

3. Configure D2-Config.properties as described in the following table:

Parameter	Description
default_language	Type the two-letter language code to set the default language and prevent users from changing their language option.

Parameter	Description
forceServerInDocbaseName	Set to `true` to force connections to use the *<repository>*@*<server>* address structure. **Note:** When forceServerInDocbaseName is set to `true`, a `.(dot)` character is allowed in Validator.HTTPCookieValue of the `ESAPI.properties`.
hideDomain	Set to `true` to hide the domain on the login dialog box. You can also specify the repository by using the parameter `hideDomain.`*<repository name>*.
docbaseFilter	Type a list of repositories, separated by commas, to be hidden from the **Repository** list box when an end user logs in.
temporaryMaxFiles	Type the maximum number of files to be temporarily stored. Once the maximum is reached, the oldest files are deleted.
logLevel	Append one of the following values: • all • info • trace • debug • warn • error
logSaveMethod	Set to `true` to save all event logs from D2Methods in the **Temp** cabinet of the repository. "Configure logback.xml for Documentum CM Server" on page 76 contains more information. By default this setting is set to `false`.
OpenText Documentum Content Management (CM) Client Branch Office Caching Services	Set to `true` to enable Branch Office Caching Services if Client Branch Office Caching Services is deployed on one or more Branch Office Caching Services servers.
includeAcsServer	Set to `true` to enable Branch Office Caching Services if Client Branch Office Caching Services is deployed on the Accelerated Content Services server on Documentum CM Server.

Parameter	Description
proxyClientIpHeader	Set to `true` to put the client IP address in the header instead of the proxy IP. Use this setting when you have a proxy in your architecture, as by default the proxy replaces the client IP address with the proxy IP. For example, if disabled, you may not be able to select the correct instance of Branch Office Caching Services.

4. In order to enable the client configuration server to load various caches when it starts up, open the `D2-Config.properties` file and update the `LoadOnStartup` parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example: `LoadOnStartup=repo1,repo2`. Note that the corresponding username and password properties for each listed repository need to be set in the global registry keystore. For example,

```
LoadOnStartup.repo1.username=dmadmin1
    LoadOnStartup.repo1.password=password1
    LoadOnStartup.repo2.username=dmadmin2
    LoadOnStartup.repo2.password=password2
```

   Or, if the listed repositories all have common admin credentials,

```
LoadOnStartup.*.username=dmadmin
    LoadOnStartup.*.password=password
```

5. Set up when and how client configuration logging events occur by configuring the following elements in `logback.xml`:

```
    <file>C:\logs\D2-D2-Config.log</file>
    <append>true</append>
    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
        <level>debug</level>
    </filter>
```

   Change the path in the `file` element if you do not want to use the default location.

   Set the logging level in the `level` element found within `<root>`:

   - `off`: no logs.
   - `error`: only exceptions.
   - `warn`: non-blocking errors.
   - `info`: HTTP data.
   - `debug`: used API methods.
   - `trace`: exchanged XML.

   The logback website (http://logback.qos.ch/) contains further information on configuration settings.

6. In client configuration, navigate to **Menu > Tools > Reload D2 options** to refresh the options.

7. Return to the instructions: "Instructions for installing the client" on page 14.

8.2 Configure the user interfaces

1. Navigate to the location of your configuration files. The settings in these files are applicable to Classic View and Smart View.

   The default location is *<install path to web application server>*/webapps/D2/WEB-INF/classes

2. Configure dfc.properties:

   a. If you want to use a shared set of configurations, configure dfc.properties to refer to an existing dfc.properties.

      By default, dfc.properties contains a reference to the dfc.properties file. All settings found in the referenced dfc.properties apply. Do not remove the # as the full command is #include, and the line is not being commented out.

      #include *<install path to Documentum>*/config/dfc.properties

   b. If you want to create application-specific settings that override the shared dfc.properties, append the settings to the dfc.properties found in *<install path to web application server>*/webapps/D2/WEB-INF/classes.

      See *<install path to Documentum>*/config/dfcfull.properties for possible settings.

   c. Ensure that the dfc.properties file being used or referred to addresses the correct docbroker and port:

      dfc.docbroker.host=*<IP address of the Fully Qualified Domain Name of the docbroker host>*

      dfc.docbroker.port=*<port>*

   d. If your Documentum CM Server installation uses non-anonymous certificates, add the following lines:

      dfc.security.ssl.truststore=*<path to dfc.keystore>*

      dfc.security.ssl.truststore_password=*<password>*

3. Configure settings.properties, which includes comments that describe each setting. Consult "settings.properties settings reference" on page 229 for more information.

4. Configure rest-api-runtime.properties as needed if you intend to use Smart View client. Refer to the rest-api-runtime.properties.template in the WEB-INF/classes folder for specific documentation on settings.

5. Configure D2FS.properties for both Classic View and Smart View. The files include comments that describe each setting. Consult "D2FS.properties settings reference" on page 189for more information.

6. In order to enable the app server to load various caches when it starts up, open its D2FS.properties file and update the LoadOnStartup parameter by setting its

value to a comma-separated list of repository names for which caches should be loaded at startup time. For example: `LoadOnStartup=repo1,repo2`. Note that the corresponding username and password properties for each listed repository need to be set in the global registry keystore. For example,

```
LoadOnStartup.repo1.username=dmadmin1
    LoadOnStartup.repo1.password=password1
    LoadOnStartup.repo2.username=dmadmin2
    LoadOnStartup.repo2.password=password2
```

Or, if the listed repositories all have common admin credentials,

```
LoadOnStartup.*.username=dmadmin
    LoadOnStartup.*.password=password
```

7. Set up when and how logging events occur by configuring the following elements in `logback.xml`:

```
    <file>C:\logs\d2.log</file>
    <append>true</append>
    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
        <level>debug</level>
    </filter>
```

Change the path in the `file` element if you do not want to use the default location.

Set the logging level in the `level` element found within `<root>`:

- `off`: no logs.
- `error`: only exceptions.
- `warn`: non-blocking errors.
- `info`: HTTP data.
- `debug`: used API methods.
- `trace`: exchanged XML.

The logback website (http://logback.qos.ch/) contains further information on configuration settings.

8. Return to the instructions: "Instructions for installing the client" on page 14.

8.2.1 Smart View client-side cache service

Smart View supports a client-side data caching service to offer faster navigation for users. The cache service runs on a user's browser and assists with repetitive data requirements by reducing the number of HTTP data requests sent to the web-server when a user quickly navigates between pages.

For more information, see *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*.

8.3 Configure the Java Method Server (JMS)

1. Stop the JMS.
2. Navigate to the `APP-INF/classes` folder of the Documentum CM Server JMS.
3. Create `d2-jms.properties` if the file does not exist.
4. To configure the order that autolink and security are applied to content, add or set the line:

   ```
   forceLinkAfterSecurity = <true or false>
   ```

   Where `true` forces the client to apply Autolink rules to content before applying Security, and `false` forces the client to apply Security rules to content before applying Autolink.
5. Restart the JMS.
6. Return to the instructions: "Instructions for installing the client" on page 14.

8.4 File transfer modes

You can configure how content transfer is performed by browser clients. The client offers two options:

- WSCTF: This mode is a web socket-based solution that works with your users' browsers to perform transfer operations.
- Thin client mode: this mode is available for all supported browsers. Note that features such as native annotations, folder import, and folder export are not available because these require WSCTF.

Note: The client functions such as the various File Transfer modes require all components to be at the same "bitness" level. Transferring a 32 bit Outlook file requires 32 bit browser and 32 bit Java parity, while 64 bit Outlook requires 64 bit browser and 64 bit Java.

8.5 Set PROXY environment variables

In WSCTF mode, in Microsoft Windows environments, it is not necessary for end users to specify HTTP_PROXY or HTTPS_PROXY environment variables since they are automatically detected from browser/system settings.

In a Mac environment, end users should continue to specify HTTP_PROXY or HTTPS_PROXY environment variables.

8.6 Understand the Documentum Client Manager v2 install

For the Web Socket Content Transfer Framework (WSCTF) feature, the WSCTF add-in is a single native desktop application available for installation using an `.msi` file (`DCMAppInstaller.msi`). WSCTF can be used with both Classic View and Smart View.

8.7 Install WSCTF

There are two ways in which the Documentum Client Manager application can be installed on the target machines:

- End user installation option
- IT Administrator push option

Note: If the WSCTF connection is not established, the browser plug-in mode falls back to thin.

If WSCTF is being used on the Microsoft Edge browser, a loopback exemption is required. Follow these steps to exempt a loopback address:

1. Open a command window as an administrator.
2. Type the following command and press **ENTER**:

```
CheckNetIsolation.exe LoopbackExempt -a -n="Microsoft.
MicrosoftEdge_8wekyb3d8bbwe"
```

Note: .NET Framework of version 4.8 or above is required on the browsing machine.

8.7.1 End user installation

1. Set `wsctf` as a value for the `browser.plugin.mode` property in the `settings.properties` file. When the user logs into Classic View or Smart View, they are prompted to install the content transfer framework. If the user is using a different client experience, such as thin and `wsctf` is mentioned in the `settings.properties` file, then the user can change the Client Plugin Mode (or Add-in) to Documentum Client Manager in the user settings. This setting change prompts them to install.
2. In the **Open URL: Documentum DCTMCTF Protocol** message box, select **Open URL:Documentum DCTMCTF Protocol**. This message box appears in the Chrome and Firefox browsers. It does not appear in the Internet Explorer and Microsoft Edge browsers.
3. In the**Install the Documentum* Client Manager** prompt, click **Yes**. The `DCMAppInstaller.msi` file downloads, which will be used to install the client manager app.

4. In the **Documentum Client Manager v2 Setup** dialog box, click **Install** to run the basic installation, which installs the files in the default directory.

   Alternately, the user can click **Advanced**, which allows them to select the install directory. An option is also available to install for all users on the computer, which requires `Administrator` privileges.

5. After the app is installed, certificates are generated and installed on the computer to secure the websocket communication. If a certificate is already installed, it will be first uninstalled. In the **Root Certificate Store** message box, click **Yes**. The certificate is imported into the trust store of the operating system. In the **Security Warning** message box, click **Yes**.

Note: The following characters are reserved for URIs and must not be used in a custom `.msi` file name:

- Single quotation mark: `'`
- Double quotation mark: `"`
- Tilde: ~

8.7.2 IT administrator push

IT administrators can push the installer on end user machines using SCCM.

Notes

- In version 16.4 and 16.5 releases, when WSCTF was IT pushed, the mechanism was a per-user installation. From version 16.5.1 onwards it became per-machine installation. If you pushed WSCTF via IT push from version 16.4 or 16.5 to a system and later push WSCTF from 16.5.1 to the same system, then there is a chance that the DCMApp invocation will fail and the end user will see a yellow bar appear repeatedly.

  To remedy this, delete the registry key from the affected systems using a command similar to the following:

  ```
  reg delete "HKCU\Software\Classes\dctmctf" /f
  ```

- The information described in this document was created and verified by running on the equipment located in OpenText labs and were performed by the OpenText engineering group. Anything beyond the information in this document may be outside the scope of standard support. Contact Open Text Corporation for more information. Note that using a configuration similar to that described in this document, or any other certified configuration, does not guarantee the results documented herein. There may be network, system parameters or other variable differences in any other environment that could affect the results. For all OpenText production deployments, OpenText recommends a rigorous testing and evaluation of the specific environment and applications to ensure that there are no configuration or custom development bottlenecks present that may result in a different outcome than covered here.

1. Set `wsctf` as a value for the `browser.plugin.mode` property in the `settings.properties` file.

2. Run the `DCMAppInstaller.msi` (with Administrator privileges) from a command prompt:

   ```
   msiexec /i [Path to the installer.msi] MSIINSTALLPERUSER="{}" /qn
   ```

   Note: The `MSIINSTALLPERUSER="{}"` parameter specifies that the installation will be per machine. The `/qn` parameter specifies that the installation will occur quietly, bypassing the interface windows for installation.

8.7.3 Verify successful installation of Documentum Client Manager v2

1. *For per-user installation:*

   a. Verify the `\ContentXfer\dcm\*.*` files (WSCTF app files) are present in the configured directory (by default it is `%localappdata%\EMC`).

   b. Verify the registry `HKEY_CURRENT_USER\Software\Classes\dctmctf\shell\open\command` contains a value of the path of `DCMApp.exe` followed by `"%1"`.

   c. Verify the certificate `Documentum Client Manager` is present in the System trusted root certificates.

   d. After the new Documentum Client Manager certificate gets imported through WSCTF installation in the Windows Certificate store, Firefox needs to be restarted before it can use the certificate from the store successfully.

2. *For per-machine installation:*

   a. Verify the `\ContentXfer\dcm\*.*` files (WSCTF app files) are present in the configured directory (by default it is `C:\Program Files (x86)\OpenText`).

   b. Verify the registry `HKEY_LOCAL_MACHINE\SOFTWARE\Classes\dctmctf\shell\open\command` contains a value of the path of `DCMApp.exe` followed by `"%1"`.

   c. Verify the certificate `Documentum Client Manager` is present in the System trusted root certificates.

   d. After the new Documentum Client Manager certificate gets imported through WSCTF installation in the Windows Certificate store, Firefox needs to be restarted before it can use the certificate from the store successfully.

8.7.4 Bypass WSCTF in Smart View

A login URL parameter is available that allows an administrator to override user Documentum Client Manager (WSCTF) add-on settings in Smart View and default to thin client mode at login time. This override facilitates the use of automation scripts for cloud deployment and testing where WSCTF prompts might interrupt the script running.

Append `?plugin_mode=thin` to login URLs to activate the override.

Example usage: `http://<domain>/D2-Smartview/ui/?plugin_mode=thin`

8.7.5 Install company-owned certificates

Before installing custom generated certificates you must add the custom generated `.crt` file to *Trusted Root Certification Authorities*. Then proceed with the WSCTF installation.

Pass the following parameters to install company owned certificates:

- *INSTALLCERT* – Set the value to "YES" to direct the installer to install the default DCM certificate pair (`ctf.pem` and `dctmctf_CA.crt`). Set the value to "NO" to prevent the install of the certificates.

  Example:

  ```
  INSTALLCERT="YES"
  ```

- *PEMPATH* – Specifies the file path of custom generated certificates. The default value of *PEMPATH* is "NO".

  Example:

  ```
  PEMPATH="C:/opentext/custom/certificates/"
  ```

Example 8-1: An example command to install custom generated certificates:

```
msiexec /i DCMAppInstaller.msi MSIINSTALLPERUSER="{}" INSTALLCERT="NO" PEMPATH="[custom
certificates path]" /qn
```

8.7.6 Uninstall Documentum Client Manager v2

On successful installation, **Documentum Client Manager v2** appears in **Add or Remove Programs** with the correct version. When the uninstallation process finishes, the registries, certificate, and files mentioned in this section that are related to the Documentum Client Manager are removed.

8.8 Configure file transfer modes

1. Navigate to `<install path to web application server>/webapps/D2/WEB-INF/classes`
2. Open `settings.properties` in a text editor and set the following line:

   `browser.plugin.mode=<modes>`

   Where you can set <*modes*> to one of the following:

browser.plugin.mode	Behavior
`thin`	Always use thin client mode for all users.
`wsctf, thin`	Default to wsctf mode for all users. Fallback to thin client mode if the browser supports wsctf mode but cannot load respective plug-in. Users can choose to use wsctf or thin client mode from the user settings dialog when configured modes are supported by browser.

3. If you upgraded the Accelerated Content Services or Branch Office Caching Services server to a version that supports content transfer in thin client mode or using direct URL, open `D2FS.properties` in a text editor and set the following line:

   `allowThinClientDirectBocsDownload=<true>`

   The default parameter value of `<false>` disables direct downloads using an Accelerated Content Services or Branch Office Caching Services server when client is in thin client mode.

4. Return to the instructions: "Instructions for installing the client" on page 14.

8.9 Configure logback.xml for Documentum CM Server

Configure logging for:

- Documentum CM Server, using the `logback.xml` located in the install path.
- The Java Method Server using the `logback.xml` located in *<install path of JMS>*`/Dctm-Server_MethodServer/deploy/Server-Apps.ear/`

  If you are using Documentum CM Server 7.1 or later, the file is located in *<install path of JMS>*`/Dctm-Server_MethodServer/deployments/Server-Apps.ear/`

1. If the JMS file is named `logback_jms_full.xml`, rename it to `logback.xml`.
2. To change when and how JMS logging events occur, configure the following elements:

   ```
   <file>C:\logs\D2-JMS.log</file>
   <append>true</append>
   <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
       <level>debug</level>
   </filter>
   ```

   Change the path in the `file` element if you do not want to use the default location.

   Set the logging level in the `level` element found within `<root>`:

   - `off`: no logs.
   - `error`: only exceptions.
   - `warn`: non-blocking errors.
   - `info`: HTTP data.
   - `debug`: used API methods.
   - `trace`: exchanged XML.

   The logback website (http://logback.qos.ch/) contains further information on configuration settings.
3. Return to the instructions: "Instructions for installing the client" on page 14.

8.10 Configure Documentum CM Server server.ini

1. On the Documentum CM Server and JMS machine, navigate to and open `server.ini` as described in the following table:

Operating system	Path
Microsoft Windows	*<install path of Documentum>*`dba\config\`*<repository name>*
A Linux environment	*<install path of Documentum>*`dba/config/`*<repository name>*

2. Set the value **mail_notification** to `TRUE` to enable mail notifications for queue work items or events for Documentum CM Server.

   If the parameter is missing you do not need to add it because the default value is `TRUE`.

3. If not present, add or set the line `return_top_results_row_based=false` to the `[SERVER_STARTUP]` section.

   This setting prevents repeating attributes from being returned as individual rows in lists such as advanced searches, property pages, and repository browser widgets.

   Note: Installation configures this setting automatically. Perform the above step to change the behavior post-installation. If you make changes to this setting, you must restart the docbase before the update will take effect.

4. If you are working in a clustered Documentum CM Server environment, add or set the line `upd_last_chg_time_from_db` to `TRUE` for each running Documentum CM Server. This setting specifies that all instances of Documentum CM Server in a clustered environment have timely access to all changes in group membership.

5. Restart the Documentum CM Server.

6. Return to the instructions: "Instructions for installing the client" on page 14.

8.11 Configure auditing

If you are installing or configuring audit for the first time, create a registered table to allow queries on the audit trail and the ability to read audit information related to deleted content. If you are upgrading, you do not need to perform these steps.

The dmadmin superuser account must have the permission to purge the audit.

1. On Documentum CM Server, run the following DQL query:

```
register table dm_audittrail_s (event_name string(64), user_name
string(32), time_stamp time, object_name string(255), string_1
string(200), string_2 string(200), string_3 string(200), string_4
string(200), string_5 string(200))
```

2. Modify the name and permissions of the registered table with the following DQL query:

   ```
   update dm_registered object set object_name = 'D2 Audits', set owner_
   table_permit = 1, set group_table_permit = 1, set world_table_permit = 1
   where object_name = 'dm_audittrail_s';
   ```

3. Return to the instructions: "Instructions for installing the client" on page 14.

8.12 Configure application server pooling session

1. Navigate to and open `dfc.properties`. If you want to configure the pooling session for specific applications, configure the `dfc.properties` in each application instead of the shared `dfc.properties`, usually found in the client folder.

2. To configure a pooling session on the application server, add or change the following lines:

   `dfc.session.pool.enable = <true or false>`

   `dfc.session.pool.expiration_interval = <duration>`

   Set the `enable` value to *<true>* to enable and `false` to disable session pools.

   Type the `expiration_interval` as the duration in seconds with a maximum value of 300. When a session has lasted this duration, it stops and starts again.

   For example, you can set *<duration>* to 300 for a duration of 5 minutes.

3. Return to the instructions: "Instructions for installing the client" on page 14 .

8.13 Configure D2EventSenderMailMethod

When the client is installed into a repository, `D2EventSenderMailMethod` updates the `mail_method` attribute of `dm_server_config` to capture and process related events. If the event is not related, the client uses the `dm_event_sender` dmbasic method.

Note: The client creates tasks internally using `D2EventSenderMailMethod` in the case of dynamic Workflows. Disabling the `mail_notification` in the `server.ini` of Documentum CM Server will effect the creation of Tasks.

If you want to use the `dm_event_sender_java` method instead of `dm_event_sender`, for example to enable multi-byte characters in messages, use Documentum Administrator to set `dm_event_sender` to use the same values as `dm_event_sender_java` as described in the following table:

Attribute	Original value	New value
method_verb	.\dmbasic.exe -f. \dm_event_sender.ebs -eMail	com.emc.documentum.server.method.eventsender.EventSender
method_type	dmbasic	java

8.14 Configure to use CTS for fast web-enabled PDF configuration renditions

To set up the CTS client environment so that linearized PDFs can be generated, follow the steps below:

1. Create a folder (for example, `realtimeclient_config`) somewhere in the app-server host and JMS host file system, then set up the following structure within the directory:

   a. Copy `aek.key` from CTS host (from `%CTS%\config`) to the folder. This will be used in `preferences.xml` as part of `<AekFilePath>/`

   b. Copy `mspassword.txt` from CTS (from `%CTS%\docbases\<your_docbase>\config\pfile`) to the `pfile` folder. This will be used in `preferences.xml` as `ServerProperty passwordFile`.

   c. Create an empty `cache` folder. This will be used in `preferences.xml` as `ServerProperty Cache`.

   d. Create a `preferences.xml` file. A sample version of this file suitable for editing can be copied from the CTS server at `%CTS%\config`. Make sure you update the file with relevant `path/docbase` values for the `ServerProperty Key` cache value, `AekFilePath`, `LoginContext DocbaseName`, `ServerProperty Key` administrator value, and `ServerProperty Key` password file value (values that must be updated in the file are shown below):

   - `<ServerProperty Key="Cache" Description="The Temporary Cache Directory" Value="C:/realtimeclient_config /cache"/>`
   - `<AekFilePath>C:/realtimeclient_config/aek.key</AekFilePath>`
   - `<LoginContext DocbaseName="DocbaseNameOfUser">`
   - `<ServerProperty Key="userName" Value="Administrator"/>`
   - `<ServerProperty Key="passwordFile" Value="C:/realtimeclient_config /pfile/mspassword.txt"/>`

   Note: In case of multiple repositories configured with CTS, you might need to add multiple login context nodes under the **<Repositories>** node in the `preferences.xml` file. Also you will have to create a separate `pfile` folder for `mspassword.txt` for each repository.

   Example: If you have two docbases, “repo1” and “repo2” configured with your CTS, you should create relevant folders.

   Also, copy the `mspassword.txt` file from the corresponding docbase folder on the CTS machine.

2. Create an environment variable called `CTS_CONFIG_LOC` with the path of the root folder that contains the `preferences.xml` file as the value. The value would be the path of the `realtimeclient_config` folder: `C:\realtimeclient_config`

3. Check the **Fast Web Compatibility** checkbox in client configuration's**C2** menu selections **Rendition configuration**, **Print configuration**, **Export configuration**, or **View configuration** to turn on linearized PDF generation. In Advanced Publishing, the checkbox can be found on the **PDF configuration** tab.

4. If you have more than one app server, repeat step 1 and step 2 in every app-server host and Java Method Server host VM.

8.15 Enable POP3S and IMAP mail configuration

8.15.1 Enable POP3S mail configuration

Follow these steps to enable POP3S mail configuration support:

1. In client configuration under **Tools > Email** ensure the **Use SSL** check box is checked.

2. Create a file named `pop_c6.properties` with the following text:

```
mail.pop3s.auth=true
mail.store.protocol=pop3s
mail.host=outlook.office365.com
mail.pop3s.port=995
```

3. On the content server, place the `pop_c6.properties` in the following location:

   `<installpath>\DctmServer_MethodServer\deployments\ServerApps.ear\APP-INF\classes`

4. Modify the `jboss-deployment-structure.xml`. The xml is located in `<installpath>\DctmServer_methodServer\ServerApps.ear\META-INF\jboss-deployment-structure.xml`. Add the following under `<dependencies>`:

```
<system export="true">
               <paths>
                   <path name="com/sun/net/ssl/internal/ssl" />
                   <path name="com/sun/net/ssl" />
               </paths>
           </system>
```

5. Edit `Module.xml` in `<installpath>\modules\system\layers\base\sun\jdk\main`. Add the following to the xml:

```
<path name="com/sun/net/ssl/internal/ssl" />
 <path name="com/sun/net/ssl" />
```

6. Edit `MailModule.xml` in `<installpath>\modules\system\layers\base\javax\mail\api\main\module.xml`. Add the following to the xml:

```
<!--<resource-root path="javax.mail-1.5.3.jar"/>-->    <resource-root
path="mail.jar"
```

7. Make JCE Enabled. Copy `US_export_policy.jar` and `local_policy.jar` to `%JAVA_HOME%\jre\lib\security`

8. Import Certificate. Copy DigiCert, DigiCertCloudCert and Office365 Cert from `C:\`, then run the following command:

Notes

- If you receive a message that the certificate is already installed, you can skip this step.

```
keytool -importcert -file C:\DigiCert.cer -trustcacerts -alias DigiCert -
keystore "D:\Documentum\Java\jre1.8.0_131\lib\security\cacerts" -storepass
changeit

keytool -importcert -file C:\DigiCertCloudCert.cer -trustcacerts -alias
DigiCertCloudCert -keystore "D:\Documentum\Java\jre1.8.0_131\lib\security
\cacerts" -storepass changeit

keytool -importcert -file C:\Office365.cer -trustcacerts -alias Office365 -
keystore  "D:\Documentum\Java\jre1.8.0_131\lib\security\cacerts" -storepass
changeit
```

- Test the implementation with a DQL query or run the following job:

```
execute do_method WITH method = 'D2WFReceiveTaskMailMethod', SAVE_RESULTS =
true,
ARGUMENTS = '-docbase_name ContentRepo01.ContentRepo01 -user_name dmadminq
 -job_id 081c396680003583 -method_trace_level 5'
```

8.15.2 Enable IMAP mail configuration

To enable IMAP email configuration:

1. In **D2 > Email configuration**, in the **Email reception server** section, select `imap` from the **Protocol** list.
2. Update the other fields, as required.
3. Click **Save**.

8.16 Enable SMTP mail configuration

You can configure the Send Mail feature so that it uses a different email than the user's email address.

- For an SSL mail server, you can configure `smtp_c6.properties` with an email address that overwrites the user's email address and what is configured in **D2 Config > From Address**.
- For a TLS mail server, specify the email address you want to use in **D2 Config > From Address** and in the `D2FS.properties` file, set *`fromAddressOfLoggedInUser`* to `false`.

Notes

- You can only use the configurations described below when deciding on whether to send email via TLS (port 587) or SSL (port 465). The configuration options in the smtp_c6.properties file must be identical across the application server as well as the Documentum CM Server JMS server.
- For SSL, ensure that the `smtp_c6.properties` file is placed at the following locations in the Application Server and the Documentum CM Server JMS:

```
Path=<Tomcat>\webapps\D2\WEB-INF\classes
Path=<Tomcat Java Method Server>\webapps\DmMethods\WEB-INF\classes
```

- For Microsoft Office 365, both the SMTP session *Login* and the mail sender's *From_Address* must match. This means, in client configuration *Login* and *From_Address* must match and *fromAddressOfLoggedInUser* must be set to false in D2FS.properties.

8.16.1 TLS configuration

This is the working configuration for Send Mail via Microsoft Office 365.

- In the D2-email_config object, select **TLS Enabled**.

8.16.2 SSL configuration

This is the working configuration for Send Mail via non-Office 365 account using SSL at smtp.port 587 and socketFactory.port 465.

1. In the D2-email_config object, deselect **TLS Enabled**.
2. In the smtp_c6.properties file deployed on both the Application Server and JMS server, the following configurations are required:

```
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.port=465
mail.smtp.socketFactory.fallback=true
```

8.17 Enable OAuth2 mail configuration (Microsoft® 365™ only)

1. In client configuration, go to **Tools > Email**, select **oauth2** as the **Protocol**, then enter the relevant information. For more information, see Section 8.2 "Configuring the Mail Server" in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*.
2. Stop Apache Tomcat.
3. Run the following command to generate certificates:

```
keytool -J-Dhttps.proxyHost=10.194.10.21 -J-Dhttps.proxyPort=3128 -printcert -rfc -
sslserver graph.microsoft.com > graphapi-cert.cer
```

   graphapi-cert.cer is generated.

   Two certificates are generated. Keep these separate with the following names: graphapi-cert1.cer and graphapi-cert2.cer.

4. Run the following command:

```
keytool -printcert -rfc -sslserver login.microsoftonline.com:443 -J-
Dhttps.proxyHost=10.194.10.21 -J-Dhttps.proxyPort=3128 > loginapi-cert.cer
```

   loginapi-cert.cer is generated. Two certificates are generated. Keep these separate with the following names: loginapi-cert1.cer and loginapi-cert2.cer.

5. Go to `C:\Documentum\dba\secure`. Run the following command to check the number of certificates in `dfc.keystore`:

```
keytool -v -list -keystore dfc.keystore
```

6. To add the certificates to `dfc.keystore`, run the following commands:

```
keytool -importcert -file graphapi-cert1.cer -keystore dfc.keystore -storepass
Password@1234567 -alias graphapi-cert1

keytool -importcert -file graphapi-cert2.cer -keystore dfc.keystore -storepass
Password@1234567 -alias graphapi-cert2

keytool -importcert -file loginapi-cert1.cer -keystore dfc.keystore -storepass
Password@1234567 -alias loginapi-cert1

keytool -importcert -file loginapi-cert2.cer -keystore dfc.keystore -storepass
Password@1234567 -alias loginapi-cert2
```

7. Restart Apache Tomcat.

Important

When upgrading the client to version 25.2 or later, run the following idql commands *before* adding the certificates:

```
alter type 'd2_mail_config' add oauth2_keys string(244) repeating, oauth2_values
string(244) repeating publish
update d2_mail_config objects set oauth2_keys[0]='oauth2_proxy_host' where r_object_id =
'<r_object_id>'
update d2_mail_config objects set oauth2_keys[1]='oauth2_proxy_port' where r_object_id =
'<r_object_id>'
update d2_mail_config objects set oauth2_keys[2]='oauth2_tenant_id' where r_object_id =
'<r_object_id>'
update d2_mail_config objects set oauth2_keys[3]='oauth2_client_id' where r_object_id =
'<r_object_id>'
update d2_mail_config objects set oauth2_keys[4]='oauth2_client_secret' where
r_object_id = '<r_object_id>'
update d2_mail_config objects set oauth2_keys[5]='oauth2_token_url' where r_object_id =
'<r_object_id>'
update d2_mail_config objects set oauth2_keys[6]='oauth2_graph_base_url' where
r_object_id = '<r_object_id>'
update d2_mail_config objects set oauth2_keys[7]='oauth2_graph_resource' where
r_object_id = '<r_object_id>'
```

8.18 Provide the online help on a local help server (Private Help Server)

The online help for this module is delivered using the OpenText Global Help Server (GHS) system, which provides your users with live access to the latest version of the help. If you cannot use the GHS system, for example, if your site does not have internet access, you can install the OpenText Private Help Server (PHS), a local version of the help system that can host your OpenText online help on your organization's network. After the PHS is installed, you can then configure your OpenText module(s) to forward all online help requests to your PHS. For detailed information about installing the PHS, see *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Notes

- The Private Help Server can support multiple OpenText modules. If the Private Help Server has already been installed within your organization to support another OpenText module, you can add additional OpenText module online helps to that installation.
- If you are replacing a previous PHS installation, see Section 2.5 “Updating a Private Help Server installation” in *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.
- If the server you want to use for the PHS installation cannot connect to the internet, see Section 1.1 “Deploying online help files in an environment without Internet access” in *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Once the PHS is installed or upgraded, you can use its Online Help Deployer to download online helps from the GHS system by entering the help deployment codes listed in “Help deployment codes” on page 84. For more information about using the codes, see Section 3 “Adding product online help to the Private Help Server” in *OpenText Help System - Private Help Server Administration Guide (OTHS-AGD)*.

Table 8-1: Help deployment codes

Code	Product
EDCCL250400-IGD	OpenText Documentum Content Management CE 25.4

After you install a Private Help Server and deploy the My Product online help, you must redirect help requests to it. This is configured in client configuration in the Smart View Help menu entry. See the *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)* for more information.

8.19 CORS configuration

The client does not support the configuration of Cross-Origin Resource Sharing (CORS) settings by default. However you can configure CORS settings according to your requirements, during deployment time at the server level.

For more information on configuring CORS settings see https://tomcat.apache.org/tomcat-9.0-doc/config/filter.html#CORS_Filter

Chapter 9

Best practices

9.1 Enable compression at the application server when using Apache Tomcat

After deploying on Apache Tomcat, you can enable compression to reduce the data transferred between the server and clients. This setting improves end-to-end response time, especially under WAN conditions, and reduces the throughput (bytes/second) for the same transaction rate.

However, the compression settings on the application server do not apply to content transfer scenarios such as importing or exporting large documents (e.g., 300 MB). For these operations, a dedicated built-in compression mechanism is used for uploads and downloads.

1. Navigate to and open *<TOMCAT_HOME>*`/conf/server.xml`
2. Configure the threshold of the content size and the type of content to be compressed:

```
<Connector port="<server port>" protocol="HTTP/1.1"
          connectionTimeout="<timeout in milliseconds>"
          redirectPort="<redirection port>"
          socketBuffer="<buffer size in bytes>"
          maxThreads="<maximum number of threads>"
          compression="<on or off>"
          compressableMimeType="text/html,text/xml,text/plain,text/javascript,text/
css,application/json"
          compressionMinSize="<files larger than this size in bytes are compressed>"
/>
```

Note: To prevent the server version from being exposed in a verbose Tomcat error, you must hide the Tomcat stack trace. To do so, add the following line under `AccessLogValve` in the `host` section of the *<apche-tomcat>*`\conf\server.xml` file:

```
<Valve className="org.apache.catalina.valves.ErrorReportValve" showReport="false"
showServerInfo="false"/>
```

9.2 Optimize performance for widgets and large numbers of content

An end user can experience some performance overhead when a large number of content is loaded into widgets, such as the Doclist, List assistance dialog, and Repository Browser.

You can limit the number of content loaded into a widget and use server-side filtering to avoid these performance issues.

1. Navigate to and open *<install path of Client>*/WEB-INF/classes/D2FS.properties
2. To configure the maximum result size for the User, Group, Doclist, Thumbnails, and List assistance widgets, set the following parameter. If the number of content found is larger than the threshold set with this parameter, the client shows a filter field at the top of the widget. The end user can type keywords in the filter to search for the object if it is not included in the truncated result.

   maxResultSetSize=*<number of results>*
3. To configure a maximum result size specifically for the User and Group widgets, set the following parameter:

   maxAdminWidgetResultSetSize=*<number of results>*
4. To configure a maximum result size specifically for the Thumbnails widget, set the following parameter:

   maxDocgalleryWidgetResultSetSize=*<number of results>*
5. To configure a maximum result size specifically for the dialog box shown when loading content, set the following parameter:

   maxListAssistanceResultSetSize=*<number of results>*
6. To configure the maximum result size for the Repository Browser or Taxonomy widgets, set the following parameter. The server returns all content for the two widgets because they do not use the maxResultSetSize parameter. The end user can experience performance overhead when loading a large folder tree or a complex taxonomy tree. This setting alleviates the burden on rendering and does not prevent the browser from taking a long time to parse the results.

   browser.folder.limit=*<number of folders>*

9.3 Improve content transfer performance

- Configure the Apache Tomcat NIO:

  a. Navigate to and open *<TOMCAT_HOME>*/conf/server.xml

  b. Configure the threshold of the content size and the type of content to be compressed:

```
<Connector port="<server
port>"protocol="org.apache.coyote.http11.Http11NioProtocol"
/>
```

9.4 Update cookie security settings

To ensure that cookies have the *secure* and *httpOnly* attribute, you must modify the web.xml file. To do so, open \webapps\D2\WEB-INF\web.xml and add the following lines:

```
<session-config>
    <cookie-config>
        <secure>true</secure>
        <http-only>true</http-only>
    </cookie-config>
</session-config>
```

9.5 General tuning tips

Tune the following parameters according to varied workload.

When using an Oracle application server:

- Modify the Oracle sessions and processes parameters.
- Set CURSOR_SHARING to FORCE.

On Documentum CM Server:

- Modify server.ini and set the concurrent_sessions parameter.

  Use the provided Java Virtual Machine tuning arguments as a starting point and adjust them upwards based on the conditions for each environment. You can refer to Oracle (http://www.oracle.com/) for Java Options information.

On the web application server:

- Modify the Java heap size, maximum threads, and GC policy.

Note: Java heap size values:

- Initial heap size: 512 MB
- Maximum heap size: 1024 MB

Chapter 10

Configure authentication

10.1 Configure OTDS support for Classic View and client configuration

OTDS enables identity management and SSO user authentication across OpenText products.

1. Set up Active Directory and OTDS. Refer to the documentation for those products for installation and configuration information.
2. Make sure the users from Active Directory can be synched into OTDS successfully. The following are sample steps that can be used to configure the OTDS to synch the active directory members:
    a. Open to otds-admin webapp in your browser using `http://<OTDS machine>:8080/otds-admin` and log in.
    b. Click **Partitions**, then click **Add > New Synchronized User Partition**.
    c. Specify the **Connection** information using the Active Directory hostname and port.
    d. Enter **Authentication** information.
    e. Do not change the **General**, **Server Settings**, and **Group Location** information. These areas are already populated with default values.
    f. Enter **User Location**. For example, User Locations can be `OU=engineering, DC=d2,D2=com` rather than `DC=d2,D2=com` to narrow down the sync to only members of the engineering organization.
    g. Do not change the **User Mappings**, **Group Mapping**, **Scheduler**, **Monitoring**, **Notifications/Search**, and **Extended functionality** information. These areas are already populated with default values.
    h. Click **Save**, then click **Actions** on the newly created partition, then select **View Members**. Note that the members from the Active Directory are synced into OTDS.
3. Configure SSO on Documentum CM Server (with otds-admin and otdsws services started). The following are sample steps to configure OTDS with Documentum CM Server:
    a. From the otds-admin webapp, click **Resources**, then click **Add**.
    b. Specify a **Resource Name**, but keep all other information as the default values.
    c. Under **Synchronization**, select the **User and group synchronization**, **Create users and groups**, and **Modify users and groups** checkboxes.

Choose **REST (Generic)** from the **Synchronization connector** drop-down list.

d. Enter the CS dmotdsrest URL `http://<CS>:9080/dmotdsrest`, **User Name**, and **Password**, then click **Test Connection**.

e. If needed, add a**client_capability** value of `2` so that the synced users in Documentum CM Server get minimum permissions to create and import content. This setting is not mandatory.

f. Set the **default_folder** resource attribute as `/%s` and retain default values for the other attributes. Click **Next**.

g. Keep the default values on the next screen, then click **Save**. Note at this time that the resource is not activated.

h. Configure an access role. Click **Access Roles**, then click **Actions** and select **Include Groups**. Click **OK**.

i. Click **View Access Role Details**.

j. Click **Add**, select the newly created partition, and click **Add Selected Items to Access Role**. The partition's user and group will be synced to Documentum CM Server.

k. Click the **Groups** tab and delete the default `otdsadmins@otds.admin` group and click **Save** to save the access role changes.

4. Make sure the users from OTDS can be synched into Documentum CM Server successfully. Follow these sample steps to synch users and groups to Documentum CM Server:

   a. From the otds-admin webapp, click **Resources**, then select the resource you just created. Click **Actions**, then select **Consolidate** to perform a full synchronization.

   b. Click **Consolidate**.

   c. Use iDQL to ensure that the users were synched into the `dm_user` table.

5. Test that the user can be authenticated successfully. Here are some sample steps to configure Documentum CM Server:

   a. Create the oauth2 client. Go to the otds-admin webapp, click **OAuth Clients** , and click **Add**. Enter **Client ID**, check **Confidential** which will provide a client secret at the end of client creation, sign out url and sign out method.

   b. Click **Next** until you get to the **Redirect URLs** page, add `https://otds.d2.com:8443/testcsotds/login` as the redirect URL. Click **Save**.

   c. Click **Save** to oauth client, you'll get a secret key. Save the secret key which you will use later.

   d. Add the OTDS host (for example, otds.d2.com) to the trusted site zone. In Internet Explorer: **Tools** > **Internet options**. Click the **Security** tab and add the OTDS host to the trusted site.

   e. Generate oauth2 token in otdsws

```
https://otds.d2.com:8443/otdsws/oauth2/auth?response_type
=token&clientid=test_client&clientsecret=5yZLvRsrJh4myEjijnNWrMnPzYBfGrWv&redir
```

```
ect_uri
=https://otds.d2.com:8443/testcsotds/login
```

Enter `<username>` and `<password>`, and click **OK**.

f. Turn on authentication log for docbase. Go to the Documentum CM Server Manager, **Repositories** tab, click the repository. Click **Edit Service**, add `-otrace_authentication`.

g. Authenticate through iapi.exe using the oauth token:

```
API> connect,d2repo,testuser1@d2.com,dm_otds_oauth=<oauth token>
```

Check the `C:\Documentum\dba\log\<docbase>.log`, you should see the authentication log below:

```
2018-06-07T09:30:00.755000    8636[6072]    0100303980000907
[AUTH]    Start-AuthenticateUserByOAuthToken:UserLoginName(testuser1@d2.com)
2018-06-07T09:30:00.771000    8636[6072]    0100303980000907
[AUTH]    End-AuthenticateUserByOAuthToken: 1
```

h. Authenticate using iapi.exe with the plain text dm_otds_password:

```
API> connect,d2repo,testuser1@d2.com,dm_otds_password=Password@123
```

Check the `C:\Documentum\dba\log\<docbase>.log`, you should see the authentication log below:

```
2018-06-07T09:49:45.708000    8636[6532]    010030398000090d
[AUTH]    Start-AuthenticateUserByOTDSPassword:UserLoginName(testuser1@d2.com)
2018-06-07T09:49:45.786000    8636[6532]    010030398000090d
[AUTH]    End-AuthenticateUserByOTDSPassword: 1
```

6. Set up the app server and test OTDS support. Follow these sample steps to configure the app server:

a. Check the existing `dm_server_config` and see if OTDSAuthentication is configured by validating the repeating attributes `app_server_name` and `app_server_uri` of `dm_server_config`.

Configuration entries look like the below if OTDSAuthentication is configured:

```
                  [1]: do_mail
                  [2]: OTDSAuthentication
app_server_uri    [0]: http://localhost:9080/DmMethods/servlet/DoMethod
                  [1]: http://localhost:9080/DmMail/servlet/DoMail
                  [2]: http://localhost:9080/OTDSAuthentication/servlet/
```

b. On each docbase where the OTDS fullway SSO is needed then , OTDSAuthentication should be configured through IAPI using the following commands. Make sure to restart JMS once.

```
retrieve,c,dm_server_config
append,c,l,app_server_name
OTDSAuthentication
append,c,l,app_server_uri
http://localhost:9080/OTDSAuthentication/servlet/authenticate
save,c,l
```

c. Navigate to `otdsauth.properties` from Content Server in the `C:\Documentum\tomcat10.*.*\OTDSAuthentication\WEB-INF\` classes.

1. `certificate`: copy the certificate content fetched from `{otds_host}/otdsws/rest/systemconfig/certificate_content`

Note: When copying the certificate content, do not allow line breaks. Remove the line breaks and carriage returns and configure the certificate content on a single line.

2. `auto_cert_refresh=true`. This automatically refreshes the certificates when certificate rollover happens.
3. `cert_jwks_url: {otds_host}otdsws/oauth2/jwks` service, which fetches the OTDS and supports only HTTP, e.g., `http://<IP>:<PORT>/otdsws/oauth2/jwks`.

Note: In `shiro.ini`, use HTTP only for `apiSVC` to fetch OTDS public keys, e.g., `X3-OTDS.apiSvc=http://<IP>:<port>/otdsws`.

For the OTExternalID1 mapping, make sure the below properties are updated like in `otdsauth.properties`:

```
is_hybrid=true
synced_user_login_name=sAMAccountName
```

4. Add the sso.context.validation=true in Webapps/WEB-INF/Classes/D2FS.properties.

Note: At the time of installation, the client will create some users to be used internally. These users have to be added under the System Account license in OTDS. The users are the following:

```
dm_bof_registry
$D2_INSTALL_OWNER
d2_mail_manager
d2_wf_notification_user
```

10.1.1 Configure OTDS to create a SAML authentication handler

1. Click **Authentication Handlers** on the left side of the OTDS Admin page.
2. Click **Add**.
3. In the **Authentication Handler Type** drop down list, select **SAML 2.0 Authentication Handler**.
4. In the **Authentication Handler Name** field, type a name for your handler.
5. Click **Next**, then, in the **User Partition** field, select the ADFS-Partition you created earlier.
6. Click **Next**, and enter the following **Parameters**:
   - Enter an **Identity Provider (IdP) Name**. The best practice is to set the **Identity Provider (IdP) Name** to the same value as the **Authentication Handler** name

- Upload the `metadata.xml` obtained from the ADFS configuration as the **IdP Metadata File** (`https://<ADFS_FQDN>/FederationMetadata/2007-06/FederationMetadata.xml`)
- Set the **IdP NameID Format** to unspecified as shown: `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
- **OTDS SP Endpoint** is not required for this setup and is typically used for proxy configurations.
- **Active By Default** is `true` (default) which prevents the user from landing on the OTDS Login page and selecting the Authentication Handler icon.
- Set the **XML Signature Algorithm** to `SHA-256` as shown: `http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`.
- Set **Claim 1** through **Claim 3** to match the format of the attributes within the ADFS claim rule.

7. Click **Next** and enter the following **Configuration** details:
   - Set the **Authentication principal attribute** to `mail`.
8. Save the authentication handler.
9. Validate that the authentication handler was successfully created by accessing the following URL that exports the SP (Service Provider) metadata: `https://<otds_ip>:<otds_port>/otdsws/login?SAMLMetadata=<saml_handler_name>`. If the result is blank, there is an issue with the configuration of the Authentication Handler. Set the OTDS logging to `DEBUG`, attempt to access the URL again and review the `otds.log` file for errors.

Note: In order for logout to function in Classic View, you need to share certificate or keystore information in the authentication handler

10.1.2 Configure OTDS logout for Classic View and client configuration

To use the Classic View OTDS logout feature, you'll need to make the following changes in the OTDS administration page:

1. Click **Authentication Handlers** on the left side of the OTDS Admin page.
2. Select **http.negotiate** on the right side of the page, click **Actions**, then select the **Disable** menu item to disable the http.negotiate Authentication Handler. If this handler is not disabled, the OTDS login dialog will not appear when the user attempts to login again if they have not closed the browser after performing the OTDS logout action.
3. Click**OAuth Clients** on the left side of the OTDS Admin page.
4. Select the OAuth client you are using, then click **Actions** and select **Properties**.

5. Select **Redirect URLs** and then click **Add** to add `<protocol>://<d2 webapp host>:<port>/D2/OTDSLogoutResponse.html` to the list.

   Note: Do *not* specify **Sign out URL** and **Sign out Method** for the OAuth client used by the Classic View webapp. Leave these two fields blank.

6. Select **Redirect URLs** and click **Add** to add *<protocol>*`://`*<d2 webapp host>*`:`*<port>*`/D2-Config/D2ConfigLogoutResponse.html` to the list.

7. Click **Save**.

After OTDS performs the logout successfully, it will redirect to this specific URL for the webapp to finish the logout process.

10.1.3 Configure support for Multi-Repo environment for OTDS

1. Make sure Multi-Repo shares the same AEK key. Verify the key in `Server.ini` if `crypto_keyname = Csaek` is the same for global repo and local repo.

2. Export the ticket key from global repo.

   ```
   ?,c,EXECUTE export_ticket_key WITH PASSWORD='password'
   ```

   Both Global docbase and Local docbase should have the same Login Ticket Key. For the Export the Login ticket Key from global repo and import the same in each local repo. So dmTicket generated on global repo can be validated on any local repo.

3. Import ticket to local docbase generated from step 2.

   ```
   ?,c,EXECUTE import_ticket_key WITH
   KEY_STRING='DM_ENCR_TEXT_V2=AAAAEE6hbRdIK9jsXskqMo
   +hi1kIAAAAbtuBXR8a9FpUa2G5ckp7qU72oYUR8QmWHL4DNbAVaLQiEteBJFDvJMX9pflBgOEupbxr/
   2xGXJOXIP+jXq7bzUoh4Y1DK8aT',PASSWORD='password'

   reinit,c
   ```

10.1.4 Single Sign On (SSO) authentication

The client offers authentication for OTDS, TrustedReverseProxy, and custom SSO integration.

Notes

- The TrustedReverseProxy form of SSO allows integrations with CA SiteMinder and IBM WebSEAL.
- The TrustedReverseProxy and custom login integrations can use Foundation Java API Principal Mode. Principal mode works by creating Documentum CM Server login tickets on behalf of users who have been authenticated by a trusted principal. This is referred to as the half-way solution and requires no authentication plug-ins or configuration on Documentum CM Server. However, it does require that a repository administrator login name and password be specified in the global registry keystore (also known as the keystore).

- You can accomplish custom authentication by using Foundation Java API Principal Authentication, plug-in Authentication, and Shiro implementation using:
  - X3TrustedReverseProxyHttpAuthenticationFilter
  - X3TrustHttpAuthenticationFilter

10.1.5 Configure TrustedReverseProxy for various SSO environments

You can use TrustedReverseProxy authentication for configuring SSO reverse proxy products such as CA SiteMinder and IBM WebSEAL. To configure TrustedReverseProxy:

1. Navigate to `webapps/D2/WEB-INF/classes/` and open `shiro.ini`. If `shiro.ini` does not exist, create a copy of `shiro-base.ini` and rename it to `shiro.ini`.
2. Edit the `shiro.ini` file with the details provided in Step 3 through Step 7.
3. Set the general *`TrustedReverseProxy`* parameters:

   ```
   [main]
   X3-TrustedReverseProxy = com.emc.x3.portal.server.filters.authc.
   X3TrustedReverseProxyHttpAuthenticationFilter
   X3-TrustedReverseProxy.defaultRepository = <default repository>
   #Authentication type
   /** = X3-TrustedReverseProxy
   ```

4. Determine whether to use SSO authentication for both application server and Documentum CM Server, or only for application server.

   Set the following property to TRUE to use SSO authentication for both application server and Documentum CM Server. The default value is FALSE, which means SSO authentication for application server only.

   ```
   X3-TrustedReverseProxy.endToEndSolution=true
   ```

5. Perform the steps in "Configure the shiro.ini file for interoperability with client configuration and the Method Server" on page 96 to configure the `shiro.ini` file.

Note: When logging in, you must add the client context to the end of the URL. For example: http://mycompany.com/D2.

10.1.6 Configure the shiro.ini file for custom SSO integrations

Starting with version 23.4, custom SSO integrations that use Foundation Java API Principal Mode authentication must explicitly declare in `shiro.ini` by setting `enableDFCPrincipalMode=true` for the custom SSO authentication filter.

1. Add relevant D2FS-trust.* properties to the keystore.
2. Implement custom filter class (`CustomAuthenticationFilter`, for example). Here is a sample implementation:

```
private boolean enableDFCPrincipalMode = true;
public boolean getEnableDFCPrincipalMode()
{
    return this.enableDFCPrincipalMode;
}
public void setEnableDFCPrincipalMode(boolean value)
{
    this.enableDFCPrincipalMode = value;
}
```

3. Include the following lines in `shiro.ini`:

```
X3-CUSTOM=com.emc.x3.portal.server.filters.authc.CustomAuthenticationFilter
X3-CUSTOM.enableDFCPrincipalMode=true
/** = X3-CUSTOM
```

4. Rebuild and redeploy the custom filter to the server, then restart the server.

10.1.7 Configure the shiro.ini file for interoperability with client configuration and the Method Server

When the administrator runs **Tools > Reload**options or **Tools > Refresh cache** in client configuration, client configuration or a D2Method running in JMS sends an HTTP request to each application server listed in **Tools > Options > Client URLs**. If SSO is being used, however, there is no way for client configuration or the D2Method to make a HTTP request in such a way that it will be authenticated by the SSO authentication filter. For this reason, the `shiro.ini` file should be configured so that the corresponding servlet endpoints are unprotected by the SSO authentication filter. To accomplish this, the following three lines should be added above the line that protects all other folders with the chosen type of SSO.

Although requests to these servlet end points will not be protected by the SSO authentication filter it does not matter because requests to these servlet endpoints must include an encrypted admin login ticket on the URL, and the only clients capable of creating such URLs are client configuration and the D2Method code that run in JMS.

The following servlet endpoints need to bypass SSO. The security is maintained because each of these servlet endpoints require that a valid login ticket is present in the URL.

The following entries under the `[urls]` section should always be present and never be commented out:

```
[urls]/**/servlet/ReloadOptions = anon/**/servlet/RefreshCache = anon/**/
servlet/LoadOnStartup = anon/**/servlet/GetBocsUploadUrl = anon/**/servlet/
DoOperation = anon/**/servlet/Download = anon/**/servlet/SetFile = anon/**/
servlet/Checkin = anon/**/servlet/ExtractProperties = anon
```

10.1.8 Configure shiro.ini for full-way SSO with OTDS for Classic View and client configuration

Version 23.4 and later OTDS is a full-way solution. It validates the AUTH Token in both the app server side as well as content server side. Therefore, we need to configure Shiro settings in both the app server side and the content server side.

For Classic View:

1. Navigate to `webapps/D2/WEB-INF/classes/` and open `shiro.ini`. If `shiro.ini` does not exist, create a copy of `shiro-base.ini` and rename it as `shiro.ini`.
2. Make the necessary changes by uncommenting the parameters in `shiro.ini` under the OTDS section. Add details as shown below:
   - `X3-OTDS=com.emc.x3.portal.server.filters.authc.X3OTDSAuthenticationFilter`
   - `X3-OTDS.defaultRepository=d2repo`
   - `X3-OTDS.apiSvc=https://otds.d2.com:8443/otdsws`
   - `X3-OTDS.authSvc=https://otds.d2.com:8443/otdsws`
   - `X3-OTDS.clientId=test_client`
   - `X3-OTDS.subscriptionName=subscription1`
   - `X3-OTDS.appName=Documentum D2 CE 23.4`
   - `X3-OTDS.userLoginNameMapping=oTExternalID3`
   - `X3-OTDS.forceLogOutFromOtds`
   - `X3-OTDS.resourceId`

Note: Ensure the app name specified for `X3-OTDS.appName` is listed in the `product_list.txt` file at `/otdsws/WEB-INF/classes`

For client configuration:

- Copy the following entries in `shiro-base.ini` under `/D2–Config/WEB–INF/classes` to `shiro.ini`:
  - `D2Config-OTDS=com.emc.d2.web.filters.auth.OTDSAuthenticationFilter`
  - `D2Config-OTDS.defaultRepository=testenv`

- The OTDS API service URL, which will be used to get the OTDS certificate. The OTDS shiro filter can obtain the OTDS certificate from *<X3-OTDS.apiSvc>*/rest/systemconfig/certificate_content, e.g., https://otds.d2.com:8443/otdsws/rest/systemconfig/certificate_content D2Config-OTDS.apiSvc=https://*<otds_ip>*:8443/otdsws.
- The OTDS authentication service URL, which will be used to redirect the client to OTDS to obtain the OAuth token first, e.g., D2Config-OTDS.authSvc=https://*<otds_ip>*:8443/otdsws.
- The client ID that was used when creating an OTDS OAuth client in the otds-admin web app, e.g., D2Config-OTDS.clientId=client127.
- Property used for cloud tenant environments only: D2Config-OTDS.clientId=client127.
- Property used to pass the application name to the OTDS auth page: D2Config-OTDS.appName=Documentum D2 CE 24.4
- Property used to map a valid user ID of OTDS. The default value oTExternalID3 maps to the UID (username@domain). To get it to work with the username only, update the property to oTExternalID1, e.g., D2Config-OTDS.userLoginNameMapping=oTExternalID3.
- Property that specifies whether the logout is only from the client or also from the OTDS session. By default, this property is true, and you can skip configuring it unless you want to log out from the client, e.g., D2Config-OTDS.forceLogOutFromOtds.
- Resource ID that is used to sync the users or groups in OTDS and that should be configured only if the revoke flag is enabled, e.g., D2Config-OTDS.resourceId.
- Property that specifies the default DM Ticket timeout. By default, if the value is not configured, the client sets it to 480 minutes. E.g., D2Config-OTDS.defaultDMTicketTimeout = 15.
- If the client configuration server is behind a reverse proxy, uncomment the property and set the value accordingly in order for features like content transfer and OTDS to work correctly. For example, if the reverse proxy has a DNS name or reverseproxy alias and it uses the default HTTPS port 443, and the name of the client configuration web application is *D2–Config*, the value of connection.remote.url will be https://reverseproxy/D2-Config. For example, D2Config-OTDS.m_connectionRemoteUrl.

With these settings, OTDS in client configuration 24.4 is a full-way solution. It validates the OAuth token on both the app server and content server side. As a result, you need to configure Shiro settings on the client configuration side in otds_auth.properties under OTDSAuthentication\WEB-INF\classes\otdsauth.properties of the JMS Services. This results in client configuration are no longer using the D2FS-trust user account for OTDS authentication. Configure the following properties in otds_auth.properties for authentication:

- certificate : Copy the certificate content fetched from *<otds_host>*/otdsws/rest/systemconfig/certificate_content.

– `auto_cert_refresh=true`: Automatically refreshes the certificates when certificate rollover happens.

– `cert_jwks_url`: *`<otds_host>`*`/otdsws/oauth2/jwks` service, which fetches OTDS.

Notes

- For each parameter significance, read the whole OTDS section from `shiro.ini` to understand fully.
- Verify if the `X3-OTDS.apiSvc` and `X3-OTDS.authSvc` properties in `shiro.ini` have different URLs. If the `X3-OTDS.apiSvc` URL does not have access to the `tokeninfo` endpoint, use the `X3-OTDS.authSvc` URL for `X3-OTDS.apiSvc`.
- Ensure the app name specified for `D2Config-OTDS.appName` is listed in the `product_list.txt` file at `/otdsws/WEB-INF/classes`

10.1.9 Manage secrets with Vault

If you are planning to manage client secrets with Vault by HashiCorp, one of the following procedures should be performed:

- When starting or re-starting Tomcat on-prem, or
- When you manually run utilities in the `D2/D2-Config/util` folder.

To configure Vault when starting or re-starting Tomcat on-prem:

1. Copy the `DSIS.zip` file from the Content Server build package `documentum_server_<version>_windows64_sqlserver.zip` (e.g., for version 24.4, the package name is `documentum_server_24.4_windows64_sqlserver.zip`).
2. Unzip the `DSIS.zip` file to a preferred location.
3. Before DSIS can be used, a token must be generated and stored in the `dfc.properties` and `application.properties` files.

   To generate the token, open a command prompt inside the `dsis` folder and run the following command (for Linux-based environments, use a colon (`:`)):

   ```
   java -cp .;dsis.jar;lib/* com.dctm.vault.TokenGenerator
   ```

   The generated token is a numeric string.
4. In the `dfc.properties` file:

   a. Update the *`dfc.dsis.daemon.token`* parameter with the token.
   b. Comment out the *`dfc.globalregistry.password`* and *`dfc.security.ssl.truststore_password`* entries.
   c. Uncomment the following entries:

      - `dfc.dsis.enabled=true`
      - `dfc.dsis.daemon.token=<the above generated token>`

- `dfc.dsis.daemon.url=http://localhost:8200/dsis`

5. In the `application.properties` file:

   - Update the *`dsis.dctm.kvpath`* parameter with the path. This path specifies the path to the secret within the vault.

     The secret name should be in the format `<docbase_name-D2_UTILITY_USER_PASSWORDS`. For example, if the `docbase` name is `testenv`, the secret name should be:

     ```
     testenv-D2_UTILITY_USER_PASSWORDS
     ```

The `D2_UTILITY_USER_PASSWORDS` is a constant and should always be suffixed after the `docbase` name. The client depends on it to fetch the secrets from the vault. If this naming convention is not followed, it will be unable to fetch secrets from the vault and will result in functional failures.

The **Key** should be the Install Username and the **Value** should be the Install Username's password.

Before using the vault to fetch passwords, you must create the secrets in the vault:

1. Start the DSIS Server.
2. Open a command prompt, navigate inside the DSIS folder, and run `dsis_start.bat`.

   Note: Once you receive the `Initialization successful` message, the app server hosting the client/client configuration and other apps can be started.

   The DSIS startup needs to be performed before starting up the app server hosting client/client configuration application or before running any of the client/client configuration command line utilities, such as `D2KeyStoreUtil`, `D2PrivilegeClientUtil`, *etc.*.

   The DSIS shutdown needs to be performed after starting up the app server hosting client/client configuration application (and verifying that the client/client configuration app is started successfully and accessible) or after running any of the client/client configuration command line utilities, such as D2KeyStoreUtil, D2PrivilegeClientUtil, *etc.*.

10.1.10 Configure IDP for electronic signature approvals (external signoff) with OTDS for Classic View

To achieve the electronic signature approvals with OTDS, a new feature is introduced to handle the electronic signature approvals for Property page, Lifecycle, Workflows etc. To achieve that, follow the below instructions:

1. Navigate to client configuration and under **D2->Options** add the relevant group that has to go for electronic signature approvals under 'Use IDP for electronic signature:'

   Example: Use IDP for electronic signature is set to 'approvalgroup' (here users belong to the approvalgroup can perform the external signoff, the same can be verified by going to respective docbase and perform IAPI with retrieve,c,d2_options and check whether external_sign_off_group parameter is set to approvalgroup.

2. Login to OTDS admin client and navigate to the respective OAuth Clients and add the redirect url example:

   ```
   https://d2client.dctm.net/D2/d2_external_signoff_page.jsp
   ```

10.2 Configure OTDS support for Smart View

To configure OTDS for Smart View, ensure you have performed the following steps as described in this chapter:

- Create a user in Active Directory.
- Sync users to OTDS from Active Directory.
- Configure OTDS SSO on Documentum CM Server 16.4P01 or later installed, OTDS 16.2.3 or later installed with otds-admin and otdsws services started)

1. Add the following properties to the `rest-api-runtime.properties` file at `<WebAppName>/WEB-INF/classes/rest-api-runtime.properties`.

   ```
   rest.security.auth.mode=<auth_mode>
   rest.security.realm.name=<Realm or Domain name>
   rest.security.otds.login.url =<OTDS Application Login URL>
   rest.security.sso.fallback.auth.mode=<- basic or -basic-ct>
   ```

   Note: For `rest.security.sso.fallback.auth.mode`, the default value is empty, which means the fallback authentication mode is not enabled. Once this is enabled, Client REST API would return `WWW-Authenticate` with `Basic` scheme when OTDS authentication fails. This would let the client know that it needs to show the login dialog instead of OTDS login dialog. Clients can then pass basic Authorization header with inline account credentials.

   Additionally, once this is enabled, the user would be presented with a Smart View basic authentication dialog after OTDS authentication is done successfully but fails at Documentum CM Server level as the user's OTDS account is not synched with the repository.

2. If you are using an external e-signoff IDP for user signing of Workflow and Property Pages updates, add the OTDS url for the e-signature dialog to `rest-api-runtime.properties`, as shown in this example, where `<client_id>` is the oAuth client configured in the OTDS console:

   ```
   rest.security.otds.esignoff.url=<OTDS Server URL>/login?response_type=id_token&client_id=<client_id>&prompt=login&authcontext=sign&cope=openid
   ```

3. To handle the OTDS session logout, specify the `rest.security.otds.logout.url` in `rest-api-runtime.properties`. For example:

   ```
   rest.security.otds.logout.url=<OTDS Server app>/logout?client_id=<client id>
   ```

   After the OTDS logout, the user will be redirected based on the configuration. By default it will redirect to the Smart View client signout screen. In the case of a logout URL configured in the client Options in client configuration, that will be set as the redirection URL from OTDS.

   Note: In the case of logout URL configuration in client Options, the URL specified needs to be added in the OTDS OAuth client Redirection URL list.

4. Launch Smart View.

   You are redirected to the OTDS sign-in page.

5. Add the Smart View host to the trusted site list.

10.3 Access an OTDS/SSO environment without using SSO

In some cases, you might need to log in using an inline account, skipping the OTDS/SSO login framework that you established for normal operations. To do this, use the `skip_sso=true` or `skipsso=true` parameter in the client URL. You can use either `skip_sso` or `skipsso`; they function identically.

With either `skip_sso=true` or `skipsso=true`, and passing `login=<user_name>`, the fallback login page appears to the user with an auto-filled login form (values passed from IURL). For example:

```
../D2?docbase=<repo_name_here>&login=<user_name>&skip_sso=true
```

Without user name included (login parameter value) the default login page appears and the user must enter login data and authenticate to continue.

Note: If a password is also included along with docbase and login, it is considered to be a direct login (directly authenticate the user, no login page presented).

Chapter 11

Install language packs

11.1 Install client configuration language packs (French-only)

1. Install the locale to the repository. The *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* contains future instructions on populating and publishing localized Data Dictionaries into the repository.

2. Download and Extract the `Documentum-Client-Installer-<version>.zip` file. Then, extract the `LanguagePacks.zip` file from it. After that, extract the `D2-Config.zip` file from the `LanguagePacks.zip` file to a temporary location.

3. Extract the `D2_language_Pack_<language>.zip` file. Then, extract the following files to the `<install path to the web application server>/webapps/D2-Config` folder:

   - `C2-LanguagePack_<language>.zip`
   - `D2-Bin-LanguagePack_<language>.zip`
   - `D2-Config-LanguagePack_<language>.zip`
   - `D2-InfoArchive-LanguagePack_<language>.zip`
   - `D2-RPS-LanguagePack_<language>.zip`
   - `D2-Specifications-LanguagePack_<language>.zip`
   - `D2-xECM-LanguagePack_<language>.zip`
   - `O2-LanguagePack_<language>.zip`

11.2 Install Classic View and Smart View language packs

Smart View and Classic View support Client Language Packs in these languages: French, German, Italian, Spanish, Japanese, Simplified Chinese, Dutch, Portuguese, Swedish, Arabic, and Korean.

1. Install the locale to the repository. Section 35 “Populating and publishing the data dictionary” in *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* contains more information on populating and publishing localized data dictionaries to the repository.

2. Download and extract the `Documentum-Client-Installer-<version>.zip` file. Then, extract the `LanguagePacks.zip` file from it. After that, extract the `D2-Classic.zip` or `D2-Smartview.zip` file from the `LanguagePacks.zip` file to a temporary location. This ZIP file includes all the available language packs for the client.
3. Under `D2-Classic.zip`, extract `D2_Language_Packs_<language>.zip` to a temporary location. Then, extract the following files to the `<install path to the web application server>/webapps/D2` folder:
   - `D2-LanguagePack_<language>.zip`
   - `C2-LanguagePack_<language>.zip`
   - `D2-Bin-LanguagePack_<language>.zip`
4. Under `D2-Smartview.zip`, extract `D2_Language_Packs_<language>.zip` to a temporary location. Then, extract the following file to the `<install path to the web application server>/webapps/D2-Smartview` folder:
   - `D2-Smartview-LanguagePack_<language>.zip`
5. Restart the Appserver.
6. Verify that the language pack installation is successful by running the user interfaces.

Note: Ensure that the following entry in the `rest-api-runtime.properties` file in `D2-Smartview\WEB-INF\classes` displays your intended locales in short form, separated by commas (default is en). For example, `rest.error.message.supported.locales=fr,en,ar`.

11.3 Install Admin Console language packs

Admin Console supports the following client language packs: French, German, Italian, Spanish, Japanese, Simplified Chinese, Dutch, Brazilian Portuguese, Russian, Swedish, Arabic, and Korean.

Note: Admin Console plug-ins do not support right-to-left languages, so the Admin Console Arabic language pack does not include plug-in language packs.

1. Download and extract the `admin-console-language_packs_<version>.zip` file to a temporary location. This includes the `admin-console_<language>.zip` file.
2. Extract the `admin-console_<language>.zip` file to the `<install path to the web application server>/webapps/Admin console` folder.
3. Restart the Appserver.
4. Verify that the language pack installation is successful by running the Admin Console.

5. If Admin Console had an existing configuration when you installed the language pack, check the configurations where the **Label** and **Description** values specific to a language are empty and fill the fields manually.

Note: Ensure that the following entry in the `rest-api-runtime.properties` file in `Admin console-\WEB-INF\classes` displays your intended locales in short form, separated by commas (default is en). For example, `rest.error.message.supported.locales=fr,en,ar`.

Chapter 12

Install OpenText Intelligent Viewing

For cloud deployment information, see the *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD-IGD)*.

For installation information, see the relevant installation guide:

- Microsoft Windows®: *OpenText Intelligent Viewing - Windows Install Guide (CLIVSA-IGD)*
- Linux®: *OpenText Intelligent Viewing - Linux Install Guide (CLIVSA-IGL)*

After you have completed the installation, configure OTDS for OpenText™ Intelligent Viewing. For more information, see Section 4.18.2 “Configuring OTDS for Intelligent Viewing” in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD-IGD)*.

Once you have completed deployment or installation, see Section 21 “Configuring OpenText Intelligent Viewing” in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)* for more information about the configuration of the viewer in client configuration for Smart View.

Note: OpenText Intelligent Viewing should not be used in concert with BravaCSR viewer or OpenText™ Brava!™ Enterprise viewer because markups and annotations are not cross-compatible.

Chapter 13

Install OpenText™ Brava!™

13.1 Introduction

Brava! offers simple-to-use annotation and collaboration, managing markup files within the repository and user interfaces. Users can view, annotate, and redact native files including PDFs, Microsoft® Office documents, image files, 2D CAD drawings, and even video clips.

This section provides an overview of the Brava! product, information about installation and configuration, as well as troubleshooting tips to assist administrators and integrators with advanced customizable Brava! features.

Refer to the *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)* for detailed information about configuring the Brava! widgets, and advanced configuration options. Check the *OpenText Documentum Content Management - Smart View User Guide (EDCCL-UGD)* for end user feature instructions such as viewer usage.

13.2 Installation prerequisites

Brava! is included as part of the installation, and its DAR file is available as part of the standard install procedure. To use the integration, additional installation and configuration steps are required.

Integration usage requirements:

1. Manual download and installation of Brava! Enterprise from My Support (https://support.opentext.com). You can download the required installation package there.

   Information for installing and configuring Brava! Enterprise can be found in the *OpenText Brava! Enterprise - Administration Guide (CLBRVW-ABE)* (check the *OpenText Documentum Infrastructure Certification Guide* for compatibility).

   > **Important**
   > Any installation and configuration information contained in this integration guide takes precedence over any seemingly conflicting information found in the core Brava! Enterprise documentation.

2. Manual activation and configuration of the Brava! widget, as described in the *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*, is a requirement.

13.3 Configure the Brava! installation

After completing the component installation, you will need to configure your installation for your specific environment.

13.3.1 Customize Brava! parameters (required)

Follow the steps below to set up `brava_parameters.properties` to fit your configuration.

1. The `brava_parameters.properties` file is located in the `D2.war/igc` subdirectory. For the Smart View client, the `brava_parameters.properties` are available by default in the directory `\D2-Smartview\ui\igc`.

   Note: Parameters must be edited with the correct parameter values as dictated by your specific environment and needs. See Section 37 "Brava! Client Parameter Configuration" in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)* for a complete listing of all configurable Brava! Enterprise client options.

   It should be noted that the `brava_parameters.properties` file contains parameters for setting behaviors that are unique to Brava!. Some Brava! Enterprise parameters that have no functional behavior for Brava! are not in the `brava_parameters.properties` so they are not covered.

2. Locate and edit the following properties in the file to match your environment:

   Important
   You must manually update these property values prior to using the Brava! integration.

   - **ServerHostName**: URL used by the integration servlet to communicate with BravaServer. Provide the host name and port number of the application URI where the BravaServer servlet is running.

     Note: The `ServerHostName` parameter is now optional. The Brava! Server URL is provided while creating the Brava! Viewer widget in client configuration. If the `ServerHostName` parameter is provided with a value, it will override the Brava! Server url of the Brava! viewer widget in client configuration.

3. Save and close the .properties file.

The table in the Brava! parameters chapter lists the parameters (and possible values) contained in the `brava_parameters.properties` page that you can customize for the Brava! ActiveX Client and HTML Viewer. Many of these parameters determine which features will be available to viewers using the Brava! ActiveX Client and HTML Viewer. You can choose to enable or disable a feature by setting its value to TRUE or FALSE.

Note: Smart View supports the HTML Viewer only. ActiveX viewer is not supported.

Blank values for parameters will cause the Brava! viewers to use the default value in cases were values are defined values other then TRUE or FALSE, or where a blank value has the effect of turning the parameter off. see Section 37 “Brava! Client Parameter Configuration” in *OpenText Documentum Content Management - Client Configuration Guide (EDCCL-AGD)*.

13.3.2 Update Brava! parameters/config

The Brava! configuration endpoint `http://<appserver>:<port>/D2/brava?jx=reload` can be used to automatically reload all parameters within the `brava_parameters.properties` file, providing administrators and application designers a quick method of testing changes to their Brava! configuration.

Note: This a secured URI and if you are not logged in as dmadmin/Administrator group, an authentication error appears.

This `brava?jx=reload` endpoint is secured and is not accessible without an authenticated administrative user. The user groups with permission to access the endpoint are configured via `brava_parameters.properties` parameter *`BravaConfigurationGroups`*. Only logged in users that are members of the groups defined in this parameter can access the functionality contained in the `brava?jx=reload` endpoint.

The login session details can be passed to the endpoint in one of two ways:

1. Users may append the parameter *`cuid`* to the http URL.

   Example 13-1:

   ```
   http://<appserver>:<port>/D2/brava?jx=
   reload&cuid=testrepo-1534176568344-dmadmin-2139153783
   ```

   The *`cuid`* parameter can be retrieved from any OAH message logged in the browser console of an active session.

2. Users may create a widget using client configuration to house the page when troubleshooting or customizing parameters. This widget configuration is not included in the default configuration.

   To do this, create a new “ExternalWidget” with the “Widget url” value of `brava?jx=reload`.

13.3.3 Configure the BravaCSR viewer to support Changemarks and rasters

1. Create the following folder structure in the system cabinet (this can be done with any client or in Documentum Administrator):

   `/System/BravaCSRViewer/config/ChangemarkConfig.xml`

   `/System/BravaCSRViewer/Images`

2. The `ChangemarkConfig.xml` file should be structured as shown in "BravaCSR viewer ChangemarkConfig.xml file sample" on page 233.:

3. BravaCSR annotation-related properties should be uncommented in `D2FS.properties`.

13.3.4 Configure the BravaCSR viewer to "chunk" PDFs for viewer optimization (optional)

The BravaCSR viewer supports chunking the viewing of PDF documents so, when larger documents are requested by the user, the viewing experience is faster because they don't have to wait for the entire document to load before the client UI displays the preview.

Note: In order for the BravaCSRViewer chunking feature to work properly, the Accelerated Content Services server must be available.

1. Verify that the Accelerated Content Services server is running.

2. Check if the Accelerated Content Services URLs are resolvable from the domain of the end-users. In a browser on the domain of the end-users, type the following URL to access the Accelerated Content Services server:

   `http://<ACS-server>:<port>/ACS/servlet/ACS`

   If the Accelerated Content Services server is resolvable from the domain of the end-users, the following message appears: `ACS Server is Running.`

3. In `web.xml` in `C:\Documentum\<Tomcat Java Method Server>\webapps\ACS\WEB-INF` add the following entry, then save `web.xml`:

```
<filter>
   <filter-name>CORSFilter</filter-name>
   <filter-class>com.documentum.osgi.filter.CORSFilter</filter-class>
   <init-param>
        <param-name>CORSAllowed</param-name>
        <param-value>true</param-value>
   </init-param>
   <init-param>
       <param-name>ExposedHeaders</param-name>
      <param-value>x-d2-client-type, x-d2-application-name, Location, Accept-
Ranges, Content-Encoding, Content-Disposition, Content-Length, Content-Range,
Authorization, DOCUMENTUM-CSRF-TOKEN, DOCUMENTUM-CSRF-

       HEADER-NAME</param-value>

   </init-param>
```

```
   <init-param>
      <param-name>AllowedHeaders</param-name>
     <param-value>contextid, objectid, x-xsrf-token, x-d2-client-type, x-d2-
application-name, x-d2-network-location, documentum-custom-unauth-scheme,
documentum-csrf-token</param-value>
   </init-param>
</filter>
<filter-mapping>
<filter-name>CORSFilter</filter-name>
<url-pattern>/servlet/*</url-pattern>
</filter-mapping>
<servlet>
<servlet-name>Bridge</servlet-name>
<servlet-class>com.documentum.osgi.servlet.BridgeServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>Bridge</servlet-name>
<url-pattern>/servlet/*</url-pattern>
</servlet-mapping>
```

4. Open `services.msc` from the Windows Run command box.
5. Restart the JMS.

   Note: Before testing chunking, make sure regular Accelerated Content Services uploads and downloads are working correctly in your environment.

6. Ensure that the chunking settings are configured properly in `D2FS.properties`, as shown:

```
# Flag to indicate whether page chunking should be allowed or not
# (Default = false)
bravacsrViewerAllowChunking = true
# Even when chunking is allowed, it is triggered only when either file size
#is more than "filesizeMB", or total number of pages in the file is more than
"pageCount"
# (Default = 5)
bravacsrViewerChunkingTriggerFilesizeMB = 5
# (Default = 15)
bravacsrViewerChunkingTriggerPagecount = 15
# Number of pages to download in one chunk
# (Default = 5)
bravacsrViewerChunkingChunksize = 5
#For existing documents with missing metadata to support chunking,
#we queue up a profile request just in time when the document is accessed in
bravaCSRViewer.
# (Default = true)
justInTimePDFChunkingValidation = true
```

7. In client configuration, select the **Chunking Enabled** check box in the BravaCSRViewer widget configuration. This setting works with the chunking settings in `D2FS.properties` in the following ways, which can affect the application of PDF Configuration-controlled view rules:

   - **Enable Chunking** in widget = `false`, app configuration in `D2FS.properties` (bravacsrViewerAllowChunking) is `false`.

     PDF renditions are delivered NOT chunked. No change to the PDF Configuration-controlled view.

- **Enable Chunking** in widget = `true`, app configuration in `D2FS.properties` (bravacsrViewerAllowChunking) is `false`

  PDF renditions are delivered NOT chunked. No change to the PDF Configuration-controlled view.

- **Enable Chunking** in widget = `false`, app configuration in `D2FS.properties` (bravacsrViewerAllowChunking) is `true`

  PDF renditions are delivered NOT chunked. No change to the PDF Configuration-controlled view.

- **Enable Chunking** in widget = `true`, app configuration in `D2FS.properties` (bravacsrViewerAllowChunking) is `true`.

  PDF renditions are delivered chunked, and NO PDF Configuration-controlled view is delivered to the user.

8. Install Document Image Services, which can be downloaded from the My Support (https://support.opentext.com) site as a `.dar` file. DIS provides end users with imaging-related features such as page serving and page modifications. The BravaCSRViewer uses these capabilities while performing various operations on documents. Before you install DIS, make sure you have:

   - Installed Documentum CM Server and created a repository.
   - Install Documentum Administrator and create a user with Documentum CM Server administrator or Superuser privileges. The *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* provides information on creating users.

9. Log in to Documentum Administrator and verify that the DIS `.dar` installed the following components:

   - annotationconfig.xml, located in System/Modules/ImagingServiceAnnotationPluginModules.
   - In System/Modules/SBO, verify that the installation created the following folders:

     `com.documentum.pageaware.pdf.IPdfPageHandlerService`

     `com.documentum.pageaware.tiff.ITiffPageHandlerService`

     `com.documentum.services.imaging.annotationservice.IAnnotationService`

     `com.documentum.services.imaging.pageaccess.IPageAccess`

   - `PageModificationModule`, located in `System/Modules/ImageServiceModule`

10. Enable CTS to interact with BravaCSRViewer. First, ensure CTS is installed and working as expected.

    a. On the CTS machine, navigate to the CTS profile, `CTSProfileService.xml` file located at `%CTS%\config` location.

b. Edit the file by modifying the `<ForClients>` element in the file `D2`, as shown:

```
.............\\profiles\\lightWeightSystemProfiles"

CTSProfileName="lightWeightSystemProfile"/>
<CTSServerProfile CTSProfileValue="C:\Documentum\CTS
\\docbases\\winsqlcs72
\\config\\profiles\\heavyWeightProfiles"
CTSProfileName="heavyWeightProfile"/>
<CTSServerProfile CTSProfileValue="/System/Media Server/Profiles"
CTSProfileName="lightWeightProfileFolder"/>
<CTSServerProfile CTSProfileValue="/System/Media Server/
System Profiles"
CTSProfileName="lightWeightSystemProfileFolder"/>
<CTSServerProfile CTSProfileValue="/System/Media Server/
Command Line Files"
CTSProfileName="heavyWeightProfileFolder"/>
<CTSServerProfile CTSProfileValue="C:\Documentum\CTS\
docbases\winsqlcs72\config\temp_profiles"
CTSProfileName="tempFileDir"/>
<CTSServerProfile CTSProfileValue="ProfileSchema.dtd"
CTSProfileName="lwProfileDTD"/>
<CTSServerProfile CTSProfileValue="MP_PROPERTIES.dtd"
CTSProfileName="hwProfileDTD"/>
<ForClients>D2</ForClients>
</ProfileManagerContext>
```

c. Restart CTS after D2-DAR install on Documentum CM Server.

d. Run the following DQL on the repository:

```
update dm_format object set richmedia_enabled = 1 where namein
('pdf')
```

e. Enable the client machine to support chunking by performing hostname mapping for the CS machine. Go to `drivers-etc-hosts` and provide CS ip and docbroker host name, for example:

```
10.194.48.171 CSWSFPISSL.otxlab.net CSWSFPISSL
```

13.3.5 Troubleshooting tips

- **Tip**: Your Browser must accept cookies in order for many of the Brava!-related features to work. Please ensure that your browser accepts cookies.
- **Tip**: When importing HPG/PLT files to the Docbase using Documentum Administrator, you must choose **AutoCAD Drawing** for the format for these files to view in both the ActiveX and HTML Viewers.
- **Problem**: Loading a PDF file that has been "Published to PDF" with password protection results in an error when trying to view the resulting file in Brava!.

  Solution: If clients want to publish and view password protected PDF files, these files must be converted on the client and not the server. Integrators need to set the parameter **ConvertOnClientImageFormats** (in the `brava_parameters.properties` file) to `TRUE`.
- **Problem**: Publishing to the docbase results in errors.

  Solution: When a markup containing a stamp with `%dbupdatestrings` is opened and dynamically updated, and no other changes to the markup are made, the markup is not recognized as having been updated and no save prompt displays

on close. A markup containing dynamic stamps needs to be saved to retain those updated values.

- **Problem**: When a user opens a markup layer containing bi-directional stamps, the `%dbupdatestrings` values are not retained after closing the markup.

  Solution: Ensure that there is no custom metadata required when importing documents. Custom required metadata is not currently handled by the importer.

- **Problem**: When trying to save a markup, I receive the following error message: "Failed to save markup"

  Solution: Ensure that the URL's you enter in your `brava_parameters.properties` file have the correct hostnames. These hostnames should be the same names that your end-user enters in their browser to ensure the session cookies are in sync. The parameter names that should be checked is *`D2URL`*.

- **Problem:** HPGL files will not view in the Brava! Client and cause a Job Processor exception error.

  Solution: All Formats within Documentum CM Server should have an associated file extension. If no extension is associated with an object's Format then Brava! attempts to retrieve file extension from the object's name property. If an extension cannot be identified from the object name then the file will fail to view. If a particular format produces an error, use Documentum Administrator to check to make sure a DOS extension exists for that format and add the DOS Extension if it is not present.

- **Problem:** A 500 error occurs while using Websphere

  Solution: If you are using IBM WebSphere 7.0 Fix Pack 19 (7.0.0.19) as the application server servlet engine for the Brava! Server, you might need to edit the Brava! Server's properties file to change the parameter *`check.cache.blocking.timeout=`* from the default value of `3` to a value of `4` or higher. This should only be needed if you are viewing large files of 12.5MB or larger. If viewing files in excess of 30MB, this value might need to be increased to `5`. Changing the Brava! Server timing resolves a (500 error) message cache issue (related to how WebSphere works) that can occur when launching large files in Brava! from the **Annotate** option.

  Sizes provided are our best estimates based on our testing.

- **Problem**: I am not able to get the Brava! ActiveX control to download on my Windows 7 machine.

  Solution: In IE (or Internet Options), add the Web site's URL as a Trusted Site. This should allow the Brava! Client ActiveX control to download and install on your Windows 7 system.

- **Problem**: I am not able to get the Brava! ActiveX control to download on first click with IE set with medium-high or high security enabled. It works if I hit the link twice.

  Solution: The Active X control will install after you click the IE information bar about installing the Brava! ActiveX Control. However, the screen is refreshed and the browse screen is shown. This is normal behavior of this security level and the

launching of the Brava! ActiveX control. Selecting **Annotate** after this action should result in normal Brava! operation.

- **Problem**: When I publish a file with markups to a Rendition, the markups are removed.

  Solution: Note that if a file with Markups is present and is published as a new Rendition, the markups will be disassociated with the main document unless the markups are saved with the **Save with all Versions** option enabled.

13.3.6 Automated install

Internet Explorer security settings and automated direct Brava! ActiveX client install on Windows client systems:

Provided the user has the underlying permissions and rights, the Brava! ActiveX Client can install without an administrator deploying the Brava! MSI install package.

1. Add the site to the Internet **Options > Security > Trusted Sites** list.
2. On the Trusted Sites Security level settings, set the **Zone** to Medium or Lower (assuming you're using the default settings for the zone).
3. Make sure the **Enable Protected Mode** is not checked.
4. Select **OK** in the Internet Options Control Panel and restart IE.
5. Access the Brava! website and attempt to open the file using Brava! (select **Annotate** from the menu after you are logged in). Brava! will install as it does on Windows XP.

If you have attempted to automatically install Brava! without following these steps, you might need to clean up files that were installed in the failed attempt.

To clean up a failed install:

1. Check the following locations and delete the listed files before attempting the install again:

   `C:\Users\<logged in user>\IGC`

   Delete the entire IGC directory and its contents.

2. Delete the class file:

   a. Open the Internet Options Control Panel and on the **General** tab, locate the **Browsing history** section.

   b. Select **Settings** to open the **Temporary Internet Files and History Settings** dialog.

   c. Select **View objects** to open the `Downloaded Program Files` folder in a new window.

   d. In the file list, check to see if the `BravaClientXView <ver> Class` is present. If so, delete it and then close all the open windows and control panels.

e. Restart the system and try running the installation again.

Note: A user account with very limited rights will not have the permissions needed for the Brava! Client Automatic Install process. In this case, a manual install will be needed.

13.3.7 Launch HTML Video Viewer when Accelerated Content Services/Branch Office Caching Services is enabled

To use Accelerated Content Services/Branch Office Caching Services with the integrated Brava! HTML Video Viewer (`brava_parameters.properties` flags *`UseACSIfAvailable`* and *`UseBOCSIfAvailable`* are set to TRUE), Accelerated Content Services/Branch Office Caching Services must be configured for cross-origin requests or an error will result.

To enable Accelerated Content Services/Branch Office Caching Services for cross-origin requests for Brava! Video Viewer:

1. For Accelerated Content Services, change the *`CORSAllowed`* parameter value to `true` (see example) in the `web.xml` file in the following location:

   `C:\Documentum\tomcat<version>\webapps\ACS\WEB-INF`

2. For , change the *`CORSAllowed`* parameter value to `true` (see example) in the `web.xml` file in the following location:

   `C:\Documentum\tomcat<version>\webapps\documentum-bocs-ws\WEB-INF`

3. After updating the two `web.xml` files, restart the Java Method Server.

Example 13-2:

```
<init-param>
 <param-name>CORSAllowed</param-name>
 <param-value>true</param-value>
 </init-param>
```

13.3.8 Updated DLLPath functionality

Example 13-3: Brava! Enterprise / Brava! Client use cases

Single user running Brava! Client per machine.

With persistence

- Default Install/Download Location and Default Persistence Location. (MOST COMMON)

  `DllPath` (implicitly set to) => `%USERPROFILE%\IGC\<version>\`

  `UserSettingsPath` Not Set => `%USERPROFIILE%\Application Data\IGC\Brava! Client\`

PersistUserSettings Not Set => true

- Custom Install/Download Location and Default Persistence Location

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client \<version>\`

  UserSettingsPath Not Set => `%USERPROFIILE%\Application Data\IGC\ Brava! Client\`

  PersistUserSettings Not Set => true

- Custom Install/Download Location and Custom Persistence Location

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client \<version>\`

  UserSettingsPath => `C:\Program Files\IGC\Brava! Client\<version> \Brava! Client\`

  PersistUserSettings Not Set => true

- Default Install/Download Location and Custom Persistence Location

  DllPath (implicitly set to) => `%USERPROFILE%\IGC\<version>\`

  UserSettingsPath => `C:\Program Files\IGC\Brava! Client\<version> \Brava! Client\`

  PersistUserSettings Not Set => true

Without persistence

- Default Install/Download Location

  DllPath (implicitly set to) => `%USERPROFILE%\IGC\<version>\`

  PersistUserSettings => false

- Custom Install/Download Location

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client \<version>\`

  PersistUserSettings => false

Example 13-4: Multiple users running Brava! Client per machine

With persistence

- Each User getting own directory of persistence files (i.e., Settings)

  Custom Install/Download Location and Default Persistence Location.

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client \<version>\`

  UserSettingsPath Not Set => `%USERPROFIILE%\Application Data\IGC\ Brava! Client\`

PersistUserSettings Not Set => true

- All Users share directory of persistence files (i.e., Settings)

  Custom Install/Download Location and Custom Persistence Location.

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client\<version>\`

  UserSettingsPath => `C:\Program Files\IGC\Brava! Client\<version>\Brava! Client\`

  PersistUserSettings Not Set => true

Without persistence

- Custom Install/Download Location

  DllPath (implicitly set to) => `C:\Program Files\IGC\Brava! Client\<version>\`

  PersistUserSettings = false

Notes

- Custom Install/Download Location requires msi script or custom `bravaclientinstall.bat` to install the Brava! Client pieces. The default`BravaClientXWrapper.cab` only installs into the `%USERPROFILE%\IGC\<version>\` directory.
- Settings files include: `Measure.ini`, `Reasons.ini`, and `ViewerConfig.xml`
- Install/Download files include:`BravaClientX.dll`, `BravaClientXWrapper.dll`, `BravaClientXWrapper_ENU.dll`, `BravaClientXWrapper_ENU.chm`, `Integration_ENU_7.<ver>.dll`, `search_macros.xml`, `ChangemarkConfig.xml`, `BravaClientSkin.xml`, `RedactionScripts/*.xml`, `StampImages/*.png`, etc.
- The directory specified in the *UserSettingsPath* must physically exist, for the parameter to work properly. The running Brava! Client will not create new directories for this parameter.
- Please see sample html files in the `PersistenceFilesLocation` directory for more specific examples information.

13.3.9 Brava! routing service with multiple servers to certify

For help on setting up an on-prem Routing Service test environment, see the *OpenText Brava! Enterprise - Administration Guide (CLBRVW-ABE)*.

Troubleshooting:

- Check that the Brava! routing server is up and running in the single server (where the BravaSDK is installed), multi server (where only the Brava! routing server is installed), and nginx load balancer, i.e., `http://rhel77pg118.otxlab.net:8081/status`.
- Make sure all the URLs are up and running with the routing server number 8081, e.g., `http://rhel77pg118.otxlab.net:8081/status` and `http://rhel77pg118.otxlab.net:8081/BravaServer/version`.
- nginx should be up and running.

Chapter 14

Install PDF Configuration

14.1 Deploy PDF Configuration automatically

The PDF Configuration plug-in adds the Portable Document Format (PDF) control capabilities. The following installation contains the steps for automatically deploying the DAR file and then installing the PDF Configuration plug-in library files on both the Documentum CM Server and application server:

Note: If directed, the core installer automatically deploys the PDF Configuration plug-in. If the PDF Configuration plug-in was not indicated during installation or if you want to update the PDF Configuration plug-in, follow this procedure.

1. Extract `Documentum-Client-Installer-<version>.zip`. After extraction, navigate to the extracted folder and then extract `Plugins.zip`.

   For Windows, use the `C2-Install-<version>.exe` application, and for Linux, use the `C2-Install-<version>.bin` application to launch installer and then click **Next**.

2. Select the installation path for the plug-in. Click **Next**.
3. Select **Documentum Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Documentum CM Server install owner's name. Click **Next**.
5. On the repository configuration panel , select the repositories for which you would like to install the client and then click **Next/Install**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy client configuration. Click **Install** and then click **Done**.

14.2 Deploy the PDF Configuration plug-in DAR manually

For both Windows and Linux, running the plug-in installer (.exe for Windows or .bin for Linux) will automatically deploy the DAR files.

For example, if you run `C2-Install-<version>.exe`, then `C2-API.jar`, `C2-Plugin.jar`, and `C2-DAR.dar` will be deployed, and `D2-Config.properties` will be configured automatically.

However, if `C2-DAR.dar` deployment fails, follow the below steps to deploy DAR manually:

1. Select the plug-in DAR from the installation path for the plug-in.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `C2-DAR.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Documentum CM Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the `nodmadmin.installparam` file if the Documentum CM Server installation owner is not named dmadmin, as described in step 4.

4. If the Documentum CM Server installation owner is not dmadmin:
   a. Create a file in a text editor and save it as `nodmadmin.installparam`.
   b. Add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:installparam="installparam">
<parameter key="dmadmin" value="<Administrator>"/>
</installparam:InputFile>
```

where <*Administrator*> is the name of the account owner for the installation.

c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.

5. Click **Install**.

6. Click **Recent DAR install log files** to review log files.

Chapter 15

Install Transfer Configuration

15.1 Deploy Transfer Configuration automatically

The Transfer Configuration plug-in adds the ability to manage transferring properties between the client and Microsoft Office documents.

The following installation contains the steps for automatically deploying the DAR file and then installing the Transfer Configuration plug-in library files on both the Documentum CM Server and application server:

Note: If directed, the core installer automatically deploys the Transfer Configuration plug-in. If the Transfer Configuration plug-in was not indicated during installation or if you want to update the Transfer Configuration plug-in, follow this procedure.

1. Extract `Documentum-Client-Installer-<version>.zip`. After extraction, navigate to the extracted folder and then extract `Plugins.zip`.

   For Windows, use the `O2-Install-<version>.exe` application, and for Linux use the `O2-Install-<version>.bin` application to launch installer, and then click **Next**.

2. Select the installation path for the plug-in. Click **Next**.

3. Select **Documentum Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.

4. Type the Documentum CM Server install owner's name. Click **Next**.

5. On the repository configuration panel , select the repositories for which you would like to install the client. Click **Next/Install**.

6. If you are installing the plug-in for application server, specify the application server location where you want to deploy client configuration. Click **Install** and then click **Done**.

15.2 Deploy the Transfer Configuration plug-in DAR manually

For both Windows and Linux, running the plug-in installer (.exe for Windows or .bin for Linux) will automatically deploy the DAR files.

For example, if you run `O2-Install-<version>.exe`, then `O2-API.jar`, `O2-Plugin.jar`, and `O2-DAR.dar` will be deployed, and `D2-Config.properties` will be configured automatically.

However, if `O2-DAR.dar` deployment fails, follow the below steps to deploy DAR manually:

1. Select the plug-in DAR from the installation path for the plug-in.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `O2-DAR.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Documentum CM Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the `nodmadmin.installparam` file if the Documentum CM Server installation owner is not named dmadmin, as described in Step 4.

4. If the Documentum CM Server installation owner is not dmadmin:
   a. Create a file in a text editor and save it as `nodmadmin.installparam`.
   b. Add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:installparam="installparam">
<parameter key="dmadmin" value="<Administrator>"/>
</installparam:InputFile>
```

where <*Administrator*> is the name of the account owner for the installation.

c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.

5. Click **Install**.

6. Click **Recent DAR install log files** to review log files.

Chapter 16

Install Recycle Bin

16.1 Deploy Recycle Bin automatically

The Recycle Bin plug-in adds recycling bin capabilities. The following installation contains the steps for automatically deploying the DAR file and then installing the Recycle Bin plug-in library files on both the Documentum CM Server and application server:

Note: If directed, the core installer automatically deploys the Recycle Bin plug-in. If the Recycle Bin plug-in was not indicated during the installation or if you want to update the Recycle Bin plug-in, follow this procedure.

1. Extract `Documentum-Client-Installer-<version>.zip`. After extraction, navigate to the extracted folder and then extract `Plugins.zip`.

   For Windows, use the `D2-Bin-Install-<version>.exe` application, and for Linux, use the `D2-Bin-Install-<version>.bin` application to launch installer, and then click **Next**.

2. Select the installation path for the plug-in. Click **Next**.
3. Select **Documentum Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Documentum CM Server install owner's name. Click **Next**.
5. On the repository configuration panel, select the repositories for which you would like to install the client. Click **Next/Install**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy client configuration. Click **Install** and then click **Done**.

16.2 Deploy the Recycle Bin plug-in DAR manually

For both Windows and Linux, running the plug-in installer (.exe for Windows or .bin for Linux) will automatically deploy the DAR files.

For example, if you run `D2-Bin-Install-<version>.exe`, then `D2-Bin-API.jar`, `D2-Bin-Plugin.jar`, and `D2-Bin-DAR.dar` will be deployed, and `D2-Config.properties` will be configured automatically.

However, if `D2-Bin-DAR.dar` deployment fails, follow the below steps to deploy DAR manually.

1. Select the plug-in DAR from the installation path for the plug-in.

2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `D2-Bin-DAR.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Documentum CM Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the `nodmadmin.installparam` file if the Documentum CM Server installation owner is not named dmadmin, as described in Step 4.

4. If the Documentum CM Server installation owner is not dmadmin:
   a. Create a file in a text editor and save it as `nodmadmin.installparam`.
   b. Add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.
omg.org/XMI" xmlns:installparam="installparam">
<parameter key="dmadmin" value="<Administrator>"/>
</installparam:InputFile>
```

   where <*Administrator*> is the name of the account owner for the installation.
   c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.

Chapter 17

Install OpenText Documentum Content Management (CM) Retention Policy Services

17.1 Deploy Retention Policy Services automatically

The Retention Policy Services plug-in adds the ability to consume retention policies and a set of configuration modules to client configuration. To use the retention policies, you must configure policies using Retention Policy Services. The *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* contains further information. The following installation contains the steps for automatically deploying the DAR file and then installing the Retention Policy Services plug-in library files on both the Documentum CM Server and application server:

Note: If directed, the core installer automatically deploys the Retention Policy Services plug-in. If the Retention Policy Services plug-in was not indicated during the installation or if you want to update the Retention Policy Services plug-in, follow this procedure.

1. Extract `Documentum-Client-Installer-<version>.zip`. After extraction, navigate to the extracted folder and then extract `Plugins.zip`.

   For Windows, use the `D2-RPS-Install-<version>.exe` application, and for Linux, use `D2-RPS-Install-<version>.bin` application to launch installer, and then click **Next**.

2. Select the installation path for the plug-in. Click **Next**.

3. Select **Documentum Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.

4. Type the Documentum CM Server install owner's name. Click **Next**.

5. On the repository configuration panel, select the repositories for which you would like to install the client. Click **Next/Install**.

6. If you are installing the plug-in for application server, specify the application server location where you want to deploy client configuration. Click **Install** and then click **Done**.

7. Stop JMS Services.

8. Follow the instructions on Documentum CM Server for your operating system as described in the following table:

MIcrosoft Windows	A Linux environment
"Install Retention Policy Services libraries on Microsoft Windows" on page 135	"Install Retention Policy Services libraries on a Linux environment" on page 136

17.2 Deploy the Retention Policy Services plug-in DAR manually

For both Windows and Linux, running the plug-in installer (.exe for Windows or .bin for Linux) will automatically deploy the DAR files.

When you run `D2-RPS-Install-<version>.exe` .jar files will be deployed and `D2-Config.properties` will be configured automatically. Which .jar files deploy depends on whether you are installing Retention Policy Services for Classic or Smart View:

- Classic deploys `D2-RPS-API.jar`, `D2-RPS-Plugin.jar`, and `D2-RPS-DAR.dar`.
- Smart View deploys `dctm-records-api.jar`, `dctm-rps-client.jar`, and `D2-RPS-DAR.dar`.

However, if `D2-RPS-DAR.dar` deployment fails, follow the below steps to deploy DAR manually.

1. Select the plug-in DAR from the installation path for the plug-in.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `D2-RPS.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Documentum CM Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the DAR files. Type the login and password for the owner account.
Input File	Select the `nodmadmin.installparam` file if the Documentum CM Server installation owner is not named dmadmin, as described in Step 4.

4. If the Documentum CM Server installation owner is not dmadmin:

a. Create a file in a text editor and save it as `nodmadmin.installparam`.

b. Add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>

<installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.
omg.org/XMI" xmlns:installparam="installparam">

<parameter key="dmadmin" value="<Administrator>"/>

</installparam:InputFile>
```

where <*Administrator*> is the name of the account owner for the installation.

c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.

5. Click **Install**.

6. Click **Recent DAR install log files** to review log files.

17.3 Install Retention Policy Services libraries on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On Documentum CM Server:

   a. Copy the following files to the `lib` folder depending on your Documentum CM Server version, as described below:

   - `DmcPolicyEngine.jar` from the OpenText Retention Policy Services Administrator (RPSA) webapp.
   - `DmcRecords.jar` from the OpenText Retention Policy Services Administrator webapp.
   - `DmcRps.jar` from the OpenText Retention Policy Services Administrator webapp.
   - `IDmcPolicyEngine.jar` from the OpenText Retention Policy Services Administrator webapp.
   - `IDmcRps.jar` from the OpenText Retention Policy Services Administrator webapp.
   - `IDmcRpsModules.jar` from the OpenText Retention Policy Services Administrator webapp.
   - `emc-policy-services.jar` from the Retention Policy ServicesWebServices.
   - `emc-retentionmarkup-services.jar` from the Retention Policy Services WebServices.

Documentum CM Server 25.2
*<install path of Documentum>**<tomcat>*\webapps\DmMethods\WEB-INF\lib

b. Copy `jaxb-api.jar`, `jaxb-xjc.jar`, and `jaxb-impl.jar` from *<path to client installation>*`\D2\WEB-INF\lib\` to the `\lib\` folder as described in the following table:

Documentum CM Server Version	Path
Documentum CM Server 23.4 and later	*<install path of Documentum>**<tomcat>*\webapps\DmMethods\WEB-INF\lib

2. Start the JMS Services.

Note: After applying `D2ApplyRentionMethod`, the JMS Services stop processing client lifecycle methods. Do the following:

1. Add `dfc.session.allow_trusted_login=true` in `dfc.properties` on the method server where the method server executes.
2. Approve all Foundation Java API connections in Documentum Administrator as a privileged client.

17.4 Install Retention Policy Services libraries on a Linux environment

1. On Documentum CM Server:

a. Copy the following files to the `lib` folder depending on your Documentum CM Server version, as described below:

- `DmcPolicyEngine.jar` from the OpenText Retention Policy Services Administrator (RPSA) webapp.
- `DmcRecords.jar` from the OpenText Retention Policy Services Administrator webapp.
- `DmcRps.jar` from the OpenText Retention Policy Services Administrator webapp.
- `IDmcPolicyEngine.jar` from the OpenText Retention Policy Services Administrator webapp.
- `IDmcRps.jar` from the OpenText Retention Policy Services Administrator webapp.
- `IDmcRpsModules.jar` from the OpenText Retention Policy Services Administrator webapp.
- `emc-policy-services.jar` from the Retention Policy Services WebServices.

- `emc-retentionmarkup-services.jar` from the Retention Policy Services WebServices.

Documentum CM Server 25.2
*<install path of Documentum CM>**<tomcat>*\webapps\DmMethods\WEB-INF\lib

b. Copy `jaxb-api.jar`, `jaxb-xjc.jar`, and `jaxb-impl.jar` from *<path to client installation>*`\D2\WEB-INF\lib\` to the `\lib\` as described in the following table:

Documentum CM Server Version	Path
Documentum CM Server 23.4 and later	*<install path of Documentum>**<tomcat>*\webapps\DmMethods\WEB-INF\lib

2. Start the JMS Services.

Note: After applying `D2ApplyRentionMethod`, the JMS Services stop processing client lifecycle methods. Do the following:

1. Add `dfc.session.allow_trusted_login=true` in `dfc.properties` on the method server where the method server executes.
2. Approve all Foundation Java API connections in Documentum Administrator as a privileged client.

Chapter 18

Install Client Branch Office Caching Services

18.1 Understand Branch Office Caching Services

Client Branch Office Caching Services allows the client to communicate with one or more Branch Office Caching Services or with Accelerated Content Services acting in the role of a Branch Office Caching Services system.

Branch Office Caching Services servers improve file transfer performance for users by connecting to a local server even when they are remote from Documentum CM Server. This allows the client to use Branch Office Caching Services for the checking in, importing, and requesting files.

Accelerated Content Services are installed as part of every Documentum CM Server installation and allow users to bypass the application server during the transfer of files. You can use specific configurations to have the client treat an Accelerated Content Services server as a Branch Office Caching Services server.

Client Branch Office Caching Services can transfer content either synchronously or asynchronously with OpenText Documentum Content Management (CM) Messaging Service. With asynchronous write, the upload performance over WAN is comparable to the case over LAN.

The process for transferring content to the client using Client Branch Office Caching Services is:

1. An end user checks in or imports a file.
2. The client attempts to locate a Branch Office Caching Services server for the file transfer and determines if the current network of the end user is associated with a specific Branch Office Caching Services server network location.
3. If no Branch Office Caching Services server is located or responds, the client uses a servlet on the application server as a fallback transfer mechanism.
4. If a Branch Office Caching Services server is located and responds, the client establishes a connection to the Branch Office Caching Services URL and transfers the file.
5. If the transfer is sent:
   - Synchronously, Client Branch Office Caching Services issues a SAVE command to indicate to Branch Office Caching Services that the file should be immediately saved to the Documentum CM Server file store.
   - Asynchronously, Client Branch Office Caching Services issues a PARK command to indicate to Branch Office Caching Services that the file should be first cached on the Branch Office Caching Services server and then moved

to the Documentum CM Server file store with the assistance of Messaging Service.

18.2 Install Client Branch Office Caching Services

Install Branch Office Caching Services on a dedicated host server machine that is local to a specific network location to improve file transfer between the Documentum CM Server and remote end-user locations. The Branch Office Caching Services host machine does not need Documentum CM Server nor a database installed on it. Minimally, the Branch Office Caching Services host machine must have Branch Office Caching Services. The following installation contains the steps for deploying the Client Branch Office Caching Services .war file on the Branch Office Caching Services or Accelerated Content Services server and then configuring other plug-ins to use the Branch Office Caching Services or Accelerated Content Services server.

1. Before installing:
   a. Ensure Branch Office Caching Services is installed on a dedicated server.
   b. To enable asynchronous transfers, ensure Messaging Service are installed and configured in the repository.
   c. Ensure Foundation Java API and the client are installed on the application server.
   d. Client Branch Office Caching Services Foundation Java API client needs to be set as a Privileged Client.
2. When upgrading or installing your repository, select **Global Registry.**
3. Deploy the content of `D2-BOCS.war` on the Branch Office Caching Services or Accelerated Content Services server, as described in the following table:

On the Branch Office Caching Services Server	On the Accelerated Content Services Server
For Documentum CM Server 7.1: "Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 7.1 to 21.2" on page 142 For Documentum CM Server 7.0: "Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 7.0" on page 142 For Documentum CM Server 23.4 and above, "Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 23.4 and above" on page 145	"Install Client Branch Office Caching Services on an Accelerated Content Services server" on page 145.

4. Configure the client to enable Branch Office Caching Services. "Configure the client for Branch Office Caching Services" on page 148 contains further instructions.

5. Set the network location identifier for the Client Branch Office Caching Services servers. "Branch Office Caching Services and Accelerated Content Services network locations" on page 159 contains more information and "Set the Branch Office Caching Services and Accelerated Content Services network locations" on page 160 contains instructions for setting the network location identifier parameter.

6. Configure other plug-ins to interact with Client Branch Office Caching Services as described in the following table:

plug-in	Instructions
Transfer Configuration	"Configure Transfer Configuration for Branch Office Caching Services" on page 149
PDF Configuration	"Configure PDF Configuration for Branch Office Caching Services" on page 151

7. You can configure the following optional settings:

   - "Enable Branch Office Caching Services Content Transfer with non-anonymous certificate-based SSL" on page 156 contains further instructions on enabling content transfer using Branch Office Caching Services with non-anonymous certificate-based SSL connections.
   - "Enable compression for upload and download" on page 158 contains further information and instructions on enabling file compression for upload and download through Client Branch Office Caching Services.
   - "Enable asynchronous Branch Office Caching Services write" on page 159 contains further instructions for enabling asynchronous write.

8. Verify the installation of Client Branch Office Caching Services and the status of the Branch Office Caching Services or Accelerated Content Services server. "Check Client Branch Office Caching Services installation" on page 154 contains further instructions.

18.3 Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 7.0

1. Download and extract the contents of `D2-BOCS.war` to *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deploy/`

   Ensure the extracted folder is named `D2-BOCS.war`

2. Delete the `D2-BOCS.war` file.

3. Copy all .jar files from *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deploy/bocs.ear/lib/` to *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/` except for the following files:

   - `commons-collections_`*<version>*`.jar`
   - `commons-io-`*<version>*`.jar`
   - `commons-lang-`*<version>*`.jar`
   - `spring-context-support-`*<version>*`.release.jar`

4. Delete the following files from *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/`

   - `jsr`*<version>*`_api.jar`
   - `jaxb-api.jar`
   - `stax-api-`*<version>*`.jar`

5. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.4 Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 7.1 to 21.2

1. Download and extract the contents of `D2-BOCS.war` to *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deployments/`

   Ensure the extracted folder is named `D2-BOCS.war`.

2. Delete the `D2-BOCS.war` file.

3. Copy all .jar files from *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deployments/bocs.ear/lib/` to *<installation path of Documentum>*`/`*<server version>*`/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/` except for the following files:

   - `commons-collections_`*<version>*`.jar`

- commons-io-*<version>*.jar
- commons-lang-*<version>*.jar
- spring-context-support-*<version>*.release.jar
- cxf-api-*<version>*.jar
- cxf-rt-bindings-soap-*<version>*.jar
- cxf-rt-bindings-xml-*<version>*.jar
- cxf-rt-core-*<version>*.jar
- cxf-rt-databinding-jaxb-*<version>*.jar
- cxf-rt-features-clustering-*<version>*.jar
- cxf-rt-frontend-jaxws-*<version>*.jar
- cxf-rt-frontend-simple-*<version>*.jar
- cxf-rt-transports-http-*<version>*.jar
- cxf-rt-ws-addr-*<version>*.jar
- cxf-rt-ws-policy-*<version>*.jar

4. Copy the following .jar files from C:\Documentum\{Tomcat version}\webapps\bocs\WEB-INF\lib to C:\Documentum\{Tomcat version}\webapps\D2-BOCS\WEB-INF\lib:
   - aspectjrt.jar
   - dfc.jar
5. Delete the following files from *<install path of Documentum>**<Java Method Server>*\server\DctmServer_BOCS\deployments\D2-BOCS.war\WEB-INF\lib\
   - jsr*<version>*_api.jar
   - jaxb-api.jar
   - stax-api-*<version>*.jar
6. Create a dummy file named D2-BOCS.war.dodeploy in *<install path of Documentum>**<Java Method Server>*\server\DctmServer_BOCS\deployments\
7. Copy anonymous-service-handler-chain.xml from the configs.jar folder to the acs/ws/ws/ folder as described in the following table:

Folders	Paths
Original location	*<install path of Documentum>**<Java Method Server>*\server\DctmServer_MethodServer\deployments\bocs.ear\lib\configs.jar\anonymous-service-handler-chain.xml

Folders	Paths
Destination	*<install path of Documentum>*| *<Java Method Server>*/server/DctmServer_MethodServer/deployments/bocs.ear/D2-BOCS.war/WEB-INF/lib/bocs-ws.jar/com/ documentum/acs/ws/ws/

8. Copy `authorized-service-handler-chain.xml` from the `configs.jar` folder to the `services/ws/` folder as described in the following table:

Folders	Paths
Original location	*<install path of Documentum>*| *<Java Method Server>*/server/DctmServer_MethodServer/deployments/bocs.ear/lib/configs.jar/authorized-service-handler-chain.xml
Destination	*<install path of Documentum>*| *<Java Method Server>*/server/DctmServer_MethodServer/deployments/bocs.ear/D2-BOCS.war/WEB-INF/lib/D2FS-Generated-*<version>*.jar/com/emc/d2fs/dctm/api/services/ws/

9. Navigate to *<install path of Documentum>*\| *<Java Method Server>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/classes, open `dfc.properties` in a text editor.

   a. Add the following line:

   ```
   dfc.bof.classloader.enable_extension_loader_first=false
   ```

   b. Add the following lines for docbroker/globalregistry settings:

   ```
   dfc.docbroker.host[0]=
   dfc.docbroker.port[0]=
   dfc.globalregistry.repository=
   dfc.globalregistry.username=
   dfc.globalregistry.password=
   ```

10. Create `jboss-deployment-structure.xml` and add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure>
    <deployment>
        <!-- Exclusions allow you to prevent the server from automatically
            adding some depdenencies -->
        <exclusions>
            <module name="org.slf4j" />
            <module name="org.slf4j.impl" />
            <module name="org.hibernate" />
        </exclusions>
    </deployment>
</jboss-deployment-structure>
```

11. Restart Branch Office Caching Services.
12. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.5 Install Client Branch Office Caching Services on a Branch Office Caching Services server for Documentum CM Server 23.4 and above

1. Download and extract the contents of `D2-BOCS.war` to *<install path of Documentum>*`|`*<Tomcat Server>*`/webapps/`

   Ensure the extracted folder is named `D2-BOCS`.
2. Delete the `D2-BOCS.war` file.
3. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.6 Install Client Branch Office Caching Services on an Accelerated Content Services server

By default, the client does not recognize an Accelerated Content Services server as a Branch Office Caching Services system. Perform the following steps if you want to include an Accelerated Content Services server as a Branch Office Caching Services system.

1. Download and extract the contents of `D2-BOCS.war` to *<install path of Documentum>*`|`*<Java Method Server>*`/server/DctmServer_MethodServer/deploy/acs.ear`

   Ensure the extracted folder is named `D2-BOCS.war`.
2. Delete the `D2-BOCS.war` file.
3. If you are using Documentum CM Server version 7.1:
   a. Copy `authorized-service-handler-chain.xml` from the `configs.jar` folder to the `services/ws/` folder as described in the following table:

Folders	Paths
Original location	*<install path of Documentum>*`|`*<Java Method Server>*`/server/DctmServer_MethodServer/deployments/acs.ear/lib/configs.jar/authorized-service-handler-chain.xml`

Folders	Paths
Destination	*<install path of Documentum>*/*<Java Method Server>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/D2FS-Generated-4.2.0.jar/com/emc/d2fs/dctm/api/services/ws/

b. Navigate to *<install path of Documentum>*/*<Java Method Server>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/classes/, open dfc.properties in a text editor, and add the following line:

```
dfc.bof.classloader.enable_extension_loader_first=false
```

Add the following lines for docbroker/globalregistry settings:

```
dfc.docbroker.host[0]=
dfc.docbroker.port[0]=
dfc.globalregistry.repository=
dfc.globalregistry.username=
dfc.globalregistry.password=
```

4. Enable the Client Branch Office Caching Services module:

a. Open application.xml in a text editor from the location described in the following table:

Documentum CM Server Version	Path
Documentum CM Server 7.1	*<install path of Documentum>*/*<Java Method Server>*/server/DctmServer_MethodServer/deployments/acs.ear/META-INF/
Documentum CM Server 7.0	*<install path of Documentum>*/*<Java Method Server>*/server/DctmServer_MethodServer/deploy/acs.ear/META-INF/

b. Add the following lines:

```
<module id="D2-BOCS"><web><web-uri> D2-BOCS.war</web-uri><context-root>/D2-BOCS
</context-root></web></module>
```

5. Navigate to ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\acs.ear\META-INF and Open the jboss-deployment-structure.xml. Create a sub-deployment section for D2-BOCS.war and add the exclusions.:

```
<sub-deployment name="D2-BOCS.war">
<exclusions>
      <module name="org.apache.log4j"/>
      <module name="org.slf4j"/>
      <module name="org.slf4j.impl"/>
```

```
    <module name="org.hibernate"/>
  </exclusions>
 </sub-deployment>
```

6. Restart the JMS.
7. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.6.1 Install Client Branch Office Caching Services on an Accelerated Content Services server with Apache Tomcat

1. Download and extract the contents of Client Branch Office Caching Services to `<install path of Documentum>/<Tomcat Server>/server/webapps/`. Ensure the extracted folder is named `D2-BOCS`.
2. If you are using Documentum CM Server version 21.2, navigate to `<install path of Documentum>/<Tomcat Server>/server/webapps/D2-BOCS/WEB-INF/classes/`, open `dfc.properties` in a text editor and add the following line:

   ```
   dfc.bof.classloader.enable_extension_loader_first=false
   ```

   Add the following lines for docbroker/globalregistry settings:

   ```
   dfc.docbroker.host[0]=
   dfc.docbroker.port[0]=
   dfc.globalregistry.repository=
   dfc.globalregistry.username=
   dfc.globalregistry.password=
   ```

3. If you are using Content Server lower than 23.4, copy the following .jar files from `C:\Documentum\{Tomcat version}\webapps\ACS\WEB-INF\lib` to `C:\Documentum\{Tomcat version}\webapps\D2-BOCS\WEB-INF\lib`:

   - aspectjrt jar
   - dfc.jar
   - emc-dfs-rt.jar
   - emc-dfs-tools.jar

4. Restart the JMS.
5. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.7 Configure the client for Branch Office Caching Services

1. Navigate to `D2/WEB-INF/classes/` and open `D2FS.properties`
2. Uncomment the line: `#pluginsOrder=D2-BOCS,C2,O2`
3. Uncomment and set the line: `D2-BOCS=true`
4. If Client Branch Office Caching Services is deployed on the Accelerated Content Services server running on Documentum CM Server, add the line `includeAcsServer=true`

   If you do not want to use an Accelerated Content Services server for Branch Office Caching Services purposes, set the value to `false`

   If the value is set to `true` but an Accelerated Content Services server is not configured for Branch Office Caching Services, the communication fails and the client uses the application server for file transfer.
5. Configure the following parameters:

Parameter	Description
Client Branch Office Caching Services	Set to `true` to enable Branch Office Caching Services if Client Branch Office Caching Services is deployed on one or more Branch Office Caching Services servers.
includeAcsServer	Set to `true` to enable Branch Office Caching Services if Client Branch Office Caching Services is deployed on the Accelerated Content Services server on Documentum CM Server.
minFileSizeForBocs	Set a minimal size in bytes for determining whether to use Client Branch Office Caching Services for file download and upload. Depending on the `includeAcsServer` parameter, if the file size is smaller than the `minFileSizeForBocs`, the client uses the Accelerated Content Services or a direct download.
cacheBocsUrl	Set to `true` to force the Documentum CM Server cache location to load before running any download or upload attempt. This requests a load on startup configuration. By default, this parameter is set to `false`.

6. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.8 Configure Transfer Configuration for Branch Office Caching Services

If you have Transfer Configuration version 2.1.0 or later, you can configure Transfer Configuration to use Branch Office Caching Services for file transfer by performing the following steps.

1. Add and set the value of `plugin_<x>=` to the path for `O2-Plugin.jar` in `D2-BOCS.properties`.

   The location of `D2-BOCS.properties` can differ based on the server type. The following table describes the locations:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/` For Documentum CM Server 7.1 and later, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/` For Documentum CM Server 7.0 and older, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/Classes/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/` For Documentum CM Server 7.1 and later, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/` For Documentum CM Server 7.0 and older, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/`

   Use forward slashes for the file path. For example:

   - To use an absolute path: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/plugins/O2-Plugin.jar`
   - To use a relative path: `/lib/plugins/O2-Plugin.jar`

<x> equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`

2. Copy `O2-API.jar` from the Transfer Configuration plug-in download or from the install path of your Transfer Configuration installation to the `lib` folder of each Branch Office Caching Services server. The following table describes the locations of the `lib` folders:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use:`<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/lib/` For Documentum CM Server 7.1 to 21.1, use *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/` For Documentum CM Server 7.0 and older, use *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/lib/` For Documentum CM Server 7.1 to 21.1, use *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/` For Documentum CM Server 7.0 and older, use *<install path of Documentum>*`/`*<Java Method Server>*`/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/`

3. Copy `O2-plugin.jar` from the Transfer Configuration plug-in download or from the install path of your Transfer Configuration installation to the `plugins` folder of each Branch Office Caching Services server. The following table describes the locations of the `plugins` folders:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/lib/plugins/` For Documentum CM Server 7.1 to 21.2, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/` For Documentum CM Server 7.0 and older, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS//WEB-INF/lib/plugins/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/` For Documentum CM Server 7.0 and older, use `<install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/ acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/`

4. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.9 Configure PDF Configuration for Branch Office Caching Services

If you have PDF Configuration version 2.1.0 or later, you can configure PDF Configuration to use Branch Office Caching Services for file transfer by performing the following steps.

1. Add and set the value of `plugin_<x>=` to the path for `C2-Plugin.jar` in `D2-BOCS.properties`.

   The location of `D2-BOCS.properties` can differ based on the server type. The following table describes the locations:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/` For Documentum CM Server 7.0 and older, use `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS. war/WEB-INF/Classes/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/` For Documentum CM Server 7.0 and older, use `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/`

Use forward slashes for the file path. For example:

- To use an absolute path: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/Classes/plugins/O2-Plugin.jar`
- To use a relative path: `/lib/plugins/C2-Plugin.jar`

`<x>` equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`

2. Copy `C2-API.jar` from the PDF Configuration plug-in download or from the install path of your PDF Configuration installation to the `lib` folder of each Branch Office Caching Services server. The following table describes the locations of the `lib` folders:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/lib/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/` For Documentum CM Server 7.0 and older, use `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS. war/WEB-INF/lib/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS/WEB-INF/lib/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/` For Documentum CM Server 7 and older, use `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/`

3. Copy `C2-plugin.jar` from the PDF Configuration plug-in download or from the install path of your PDF Configuration installation to the `plugins` folder of each Branch Office Caching Services server. The following table describes the locations of the `plugins` folders:

Server type	Location
Branch Office Caching Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS//WEB-INF/lib/plugins/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/` For Documentum CM Server 7.0 and older, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/`
Accelerated Content Services	For Documentum CM Server 21.2 and later, use: `<install path of Documentum>/<tomcat>/Webapps/D2-BOCS//WEB-INF/lib/plugins/` For Documentum CM Server 7.1 to 21.1, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/` For Documentum CM Server 7.0 and older, use *`<install path of Documentum>`*`/`*`<Java Method Server>`*`/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/`

4. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.10 Check Client Branch Office Caching Services installation

You can perform the following steps to confirm the state of your Branch Office Caching Services installation.

1. To check for the correct installation of Client Branch Office Caching Services on a Branch Office Caching Services server, navigate to `http://`*`<bocs_server_name>`*`:8086/D2-BOCS/` in Microsoft Internet Explorer.

   If Client Branch Office Caching Services is installed and running correctly, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
-<bocs version="4.x.x build xxx" server_time="x.xxxs">
```

```
  <plugins />
</bocs>
```

If you have plug-ins installed, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
-<bocs version="4.x.x build xxx" server_time="x.xxxs">
  <plugins>C2 v4.x.x build xx</plugins>
</bocs>
```

2. To check for the correct installation of Client Branch Office Caching Services on an Accelerated Content Services server, navigate to `http://<documentum_server_name>:9080/D2-BOCS/` in Microsoft Internet Explorer.

   If Client Branch Office Caching Services is installed and running correctly, the browser shows the following message:

   ```
   <?xml version="1.0" encoding="utf-8">
   -<bocs version="4.x.x build xxx" server_time="x.xxxs">
     <plugins />
   </bocs>
   ```

   If you have plug-ins installed, the browser shows the following message:

   ```
   <?xml version="1.0" encoding="utf-8">
   -<bocs version="4.x.x build xxx" server_time="x.xxxs">
     <plugins>C2 v4.x.x build xx</plugins>
   </bocs>
   ```

3. To check the status of your Branch Office Caching Services server, navigate to `http://<bocs_server_name>:8086/bocs/servlet/ACS`

   If the Branch Office Caching Services server is running, the browser shows the answer message `ACS Server is running`

   You can also check for the status of the Branch Office Caching Services server through the client and client configuration.

4. To check the status of your Accelerated Content Services server, navigate to `http://<documentum_server_name>:9080/ACS/servlet/ACS`

   If the Branch Office Caching Services server is running, the browser shows the answer message `ACS Server is running`

   You can also check for the status of the Branch Office Caching Services server through the client and client configuration.

5. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

Note: Branch Office Caching Services and Client Branch Office Caching Services support only forward proxy (the expectation is that both are deployed under a client subnet). Reverse proxy configuration is currently not supported.

Configuring a proxy for ACS in the Branch Office Caching Services `acs.properties` file, by adding `https.proxyHost` for Branch Office Caching Services in the system properties, enables forward proxy for the entire application server. This causes the following exception:`java.io.IOException: Unable to tunnel through proxy. Proxy returns "HTTP/1.1 403 Proxy Error"`.

To avoid this issue, add the following to the Tomcat `catalina.bat` or `catalina.sh` file:

```
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.nonProxyHosts="localhost|127.0.0.1|<BOCS Host name>|<D2
hostname>"
```

18.11 Enable Branch Office Caching Services Content Transfer with non-anonymous certificate-based SSL

1. Enable non-anonymous certificate-based Secure Sockets Layer (SSL) on your Documentum CM Server. *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* contains further information and instructions on enabling non-anonymous certificate based SSL.
2. Enable SSL on the Branch Office Caching Services and Messaging Service servers. The *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)* contains instructions for enabling SSL.
3. Copy `dfc.keystore` from Documentum CM Server to the Branch Office Caching Services server.

   For example, *<install path to Documentum>*`/dba/secure/`
4. Navigate to *<install path of web application server>*`/webapps/D2-BOCS/WEB-INF/classes` and open for editing `dfc.properties`
5. Add the following lines:

   `dfc.security.ssl.truststore=`*<path to dfc.keystore>*

   `dfc.security.ssl.truststore_password=`*<password>*
6. Configure Client Branch Office Caching Services SSL:

   - App Server configurations:
     1. Import all the certificates into cacerts (D2,bocs, acs certifcates).
     2. Update the `server.xml` file from `<TOMCAT_HOME>/conf` with the ciphers:

```
<Connector
protocol="HTTP/1.1"
port="8443"
maxThreads="150"
SSLEnabled="true" scheme="https" secure="true"
SSLProtocol="TLSv1.2">
<SSLHostConfig
ciphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_G
CM_SHA384">
<Certificate
certificateKeystoreFile="C:/newssl/bocsss1244.jks"
certificateKeystorePassword="changeit"
type="RSA"
/>
</SSLHostConfig>
</Connector>
```

3. Update the `catalina.bat` file with following properties:

```
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.debug=ssl:handshake"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.debug=all"
set "JAVA_OPTS=%JAVA_OPTS% -Djdk.tls.client.protocols=TLSv1.2"
set "JAVA_OPTS=%JAVA_OPTS% -
Dsun.security.ssl.allowUnsafeRenegotiation=true"
set "JAVA_OPTS=%JAVA_OPTS% -Dhttps.protocols=TLSv1.2"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=C:/certs/cacerts -
Djavax.net.ssl.keyStore=C:/certs/cacerts -
Djavax.net.ssl.keyStorePassword=changeit -
Djavax.net.ssl.trustStorePassword=changeit"
set "JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.security.enableAIAcaIssuers=true"
```

4. Update `dfc.properties` and set `dfc.security.ssl.use_existing_truststore=true`.
5. Update the `<JDKHOME>/conf/java`.security file and change `keystore.type=pkcs12` to `keystore.type=JKS`.

- Branch Office Caching Services server configurations:
  1. Import Branch Office Caching Services and Accelerated Content Services the certificates into cacerts (Branch Office Caching Services, Accelerated Content Services, and Messaging Service certificates).
  2. Update the `server.xml` file in `<TOMCAT_HOME>/conf` with the following changes:

```
<Connector
protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443"
maxThreads="150"
SSLEnabled="true">
<SSLHostConfig>
<Certificate
  certificateKeystoreFile="C:/Documentum/tomcat10.1.10/conf/bocs.keystore"
  certificateKeystorePassword="changeit"
  type="RSA"
  />
</SSLHostConfig>
</Connector>
```

  3. Update the `<JDKHOME>/conf/java.security` file and change `keystore.type=pkcs12` to `keystore.type=JKS`.
- Accelerated Content Services server configurations:
  1. Import Accelerated Content Services and Branch Office Caching Services the certificates into cacerts (Branch Office Caching Services, Accelerated Content Services, and Messaging Service certificates).
  2. Update the `server.xml` file in `<TOMCAT_HOME>/conf` with the following changes:

```
<Connector
protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443"
maxThreads="150"
SSLEnabled="true">
<SSLHostConfig>
<Certificate
  certificateKeystoreFile="C:/Documentum/tomcat10.1.10/ACSSSL/
acs009.keystore"
  certificateKeystorePassword="changeit"
```

```
  type="RSA"
  />
</SSLHostConfig>
</Connector>
```

3. Update the `<JDKHOME>/conf/java.security` file and change `keystore.type=pkcs12` to `keystore.type=JKS`.

- Browser client configurations:

  1. Import all the certificates (Accelerated Content Services, Branch Office Caching Services, client) into the browser client.
  2. Import all the certificates into Manage computer certificate > Trusted Root certificate > certificate.

7. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.12 Enable compression for upload and download

1. To enable compression between the client application and Client Branch Office Caching Services:

   a. Navigate to and open `D2-BOCS.war/WEB-INF/web.xml`

   b. Uncomment the lines:

```
<filter>
  <filter-name>CompressingFilter</filter-name>
  <filter-class>com.planetj.servlet.filter.compression.CompressingFilter</
filter-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>compressionThreshold</param-name>
    <param-value>1024</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>CompressingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

2. To enable compression between Client Branch Office Caching Services and Documentum CM Server for non-PDF Configuration/Transfer Configurationdeployments:

   a. Navigate to and open `/D2/WEB-INF/classes/settings.properties`

   b. Add or set the following parameters:

   `applet.download.compression.enabled = true`

   `applet.upload.compression.enabled = true`

   `applet.upload.compression threshold = 1024`

   `applet.upload.compression.extensions = doc,docx,xls,xlsx,ppt,pptx, pdf,txt`

c. Navigate to and open `/D2-BOCS/WEB-INF/classes/D2-BOCS.properties`

d. Uncomment and set the following parameter:

```
compressedExtensions = doc,docx,xls,xlsx,ppt,pptx,pdf,txt
```

3. Enable Client Branch Office Caching Services:

a. Navigate to and open `/D2/WEB-INF/classes/D2FS.properties`

b. Add or set the following parameters:

```
D2-BOCS=true
```

```
includeAcsServer=true
```

4. Return to the Branch Office Caching Services installation instructions: "Install Client Branch Office Caching Services" on page 140.

18.13 Enable asynchronous Branch Office Caching Services write

1. Log in to Documentum Administrator.
2. Navigate to **[Global Repository]** > **Administration** > **Distributed Content Configuration** > **Distributed Transfer**.
3. Open the properties of the `ContTransferConfig` object.
4. Set the `ACS Write` property to **Synchronous and Asynchronous Write**.
5. Save the object properties.

18.14 Branch Office Caching Services and Accelerated Content Services network locations

You must set the `networkLocationId` for each Client Branch Office Caching Services server based on the `Network Location Identifier` property as specified in Documentum Administrator. Setting these values ensures that the Client Branch Office Caching Services server communicates with adjacent Branch Office Caching Services or Accelerated Content Services servers rather than a remote Branch Office Caching Services or Accelerated Content Services server. If there are several network locations associated with a Branch Office Caching Services or Accelerated Content Services server configuration, you should choose the one that is referenced only by that Branch Office Caching Services or Accelerated Content Services server.

For example, if you have Client Branch Office Caching Services deployed to:

- A Branch Office Caching Services server (B1) whose network locations have the identifiers NL1 and NL2.
- A Branch Office Caching Services server (B2) whose network locations have the identifiers NL2 and NL3.

For the Client Branch Office Caching Services server on the B1 server, set `networkLocationId=NL1`, and for the Client Branch Office Caching Services server on the B2 server, set `networkLocationId=NL3`.

Accelerated Content Services servers are not typically associated with a network location. In order for failover to Client Branch Office Caching Services on an Accelerated Content Services server to work correctly, you must associate a network location with the Accelerated Content Services server. You can do this by creating a Network Location in Documentum Administrator and setting the `networkLocationId` for the Accelerated Content Services server.

For example, if you have Client Branch Office Caching Services deployed to an Accelerated Content Services server (A1) whose network location has the identifier NL0, for the Client Branch Office Caching Services server on the A1 server, set `networkLocationId=NL0`.

18.15 Set the Branch Office Caching Services and Accelerated Content Services network locations

1. Open `D2-BOCS.properties` in a text editor.

   The location of `D2-BOCS.properties` can differ based on the server type. The following table describes the locations:

Server type	Location
Branch Office Caching Services	*<install path of Documentum>*/*<tomcat>*/Webapps/D2-BOCS/WEB-INF/Classes/
Accelerated Content Services	*<install path of Documentum>*/*<tomcat>*/Webapps/D2-BOCS/WEB-INF/Classes/

2. Set the network location for the Client Branch Office Caching Services server on each Branch Office Caching Services and Accelerated Content Services server you are using.

18.16 Use Client Branch Office Caching Services for download

Download performance as tested on a 300ms latency 2Mbps bandwidth connection is compared in the following table:

Download in WAN (seconds)	Application Server	Accelerated Content Services	Downloading for the first time through Branch Office Caching Services	Downloading a second time through Branch Office Caching Services
10kb	1.3	1.3	1.3	<0.1
100kb (compression ratio 50%)	2.2	1.9	1.9	<0.1
1MB (compression ratio 50%)	3.8	3.6	3.5	0.2
10MB (compression ratio 50%)	24.5	23.8	23.7	0.8
100MB (compression ratio 50%)	225.9	219.9	221.4	6.3

The tests show that:

- There are no significant differences if you use an application server or use an Accelerated Content Services the first time content is downloaded.
- Branch Office Caching Services significantly improves performance from the second time content is downloaded onward.

To use Client Branch Office Caching Services:

1. Navigate to and open *<install path to web application server>*`/webapps/D2/WEB-INF/classes/D2FS.properties`
2. Tune the `minFileSizeForBocs` parameter to force the client to not use Branch Office Caching Services for small files when performing both download or upload operations. Use the breakeven file size where using Client Branch Office Caching Services does not have a positive or negative impact on file transfer performance.

   Conduct tests in the production environment because the optimal value depends on the network conditions. The OpenText internal test results are compared in the following table:

Download in WAN (seconds)	Application Server	Accelerated Content Services	Downloading for the first time through Client Branch Office Caching Services	Downloading a second time through Client Branch Office Caching Services
10kb	1.3	1.3	12.0	10.0
100kb (compression ratio 50%)	2.2	1.9	12.8	10.1
1MB (compression ratio 50%)	3.8	3.6	15.2	10.6
10MB (compression ratio 50%)	24.5	23.8	36.2	13.3
100MB (compression ratio 50%)	225.9	219.9	258.5	25.4

The comparison shows that Client Branch Office Caching Services:

- Significantly improves performance on subsequent downloads when downloading a roughly 5MB size file at a compression ratio of 50%.
- Introduces an overhead of 10 seconds with the first and the subsequent download as a function of WAN conditions.

3. In order to enable the Client Branch Office Caching Services server to load various caches when it starts up, open its `D2-BOCS.properties` file and update the `LoadOnStartup` parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example, `LoadOnStartup=repo1,repo2`.

   Note that the corresponding username and password properties for each listed repository need to be set in the global registry keystore. For example,

   ```
   LoadOnStartup.repo1.username=dmadmin1
       LoadOnStartup.repo1.password=password1
       LoadOnStartup.repo2.username=dmadmin2
       LoadOnStartup.repo2.password=password2
   ```

   Or, if the listed repositories all have common admin credentials,

   ```
   LoadOnStartup.*.username=dmadmin
       LoadOnStartup.*.password=password
   ```

4. To initialize an expensive precache process of Branch Office Caching Services locations on startup instead of the initial content transfer:

   a. Set the following parameters:

      `LoadOnStartup.docbase`

      `LoadOnStartup.username`

`LoadOnStartup.password`

`LoadOnStartup.domain`

b. Set the `cacheBocsUrl` parameter to `true` in the following two locations:

- *<install path to web application server>*`/webapps/D2/WEB-INF/classes/D2FS.properties`
- *<install path to D2-BOCS>*`/WEB-INF/classes/D2-BOCS.properties`

18.16.1 Workaround when Client Branch Office Caching Services download/view operation fails with Transfer Configuration or PDF Configuration configured

Issue

Client Branch Office Caching Services download operation fails when the Transfer Configuration or PDF Configuration plug-ins are configured and the following error message is displayed in the download message box:

```
HttpError in request to server:400
```

Cause

For a document, if Transfer Configuration or PDF Configuration plug-in configuration is applied while performing a download or view operation, it sends offline data for processing to the Client Branch Office Caching Services server, which eventually increases the request header size. If the request header size is greater than the server-configured maximum header size, the following exception occurs:

```
org.apache.coyote.http11.Http11Processor.service Error parsing HTTP request header
 Note: Further occurrences of HTTP request parsing errors will be logged at DEBUG
level.
        java.lang.IllegalArgumentException: Request header is too large
```

Workaround

1. Update the *maxHttpHeaderSize* value in the `<connector>` tag in the Tomcat `server.xml` file:

   ```
   maxHttpHeaderSize="1655360"
   ```

   Note: The above value (`1655360`) can be updated, depending on your requirements.

2. If SSL is enabled, set the *SSLEnabled* value to `true`.

18.17 Use Client Branch Office Caching Services for upload

Do not use Client Branch Office Caching Services for upload unless you upload large content. The OpenText internal test results for upload performance are compared in the following table:

Upload in WAN (seconds)	Application Server	Accelerated Content Services	Synchronous write through Client Branch Office Caching Services	Asynchronous write through Client Branch Office Caching Services
10kb	2.2	2.0	25.2	23.1
100kb (compression ratio 50%)	3.7	3.3	25.8	24.5
1MB (compression ratio 50%)	6.9	6.1	28.4	24.6
10MB (compression ratio 50%)	26.4	25.5	51.8	28.1
100MB (compression ratio 50%)	230.3	230.1	267.8	38.4

The comparison shows that:

- With small content, both synchronous and asynchronous write introduce an overhead of about 20 seconds as a function of WAN conditions.
- With large content (>1MB), asynchronous write has superior performance to synchronous write.
- With larger content (>10M), asynchronous write performs better than using Accelerated Content Services.

Chapter 19

Install and configure OpenText Core Share components

Configure OpenText™ Core Share to work with Smart View to share content with external OpenText Core Share users.

1. Ensure that you have followed the instructions in the *OpenText Documentum Connector for Core Share - Deployment Guide (EDCCOCDFS-IGD)*.
2. Ensure that the Client REST API .war is installed. Make sure the installer is used which would install all the required dars. Then copy the Client REST API .war in the DCC app server. This D2Rest dfc should point to same docbase as Smart View docbase.
3. Update `d2fs-rest-web/web-inf/classes/rest-api-runtime.properties.`

```
#This value should be same as given in D2-Smartview\WEB-INF\
classes\rest-api-runtime.properties:
rest.security.crypto.key.salt=
rest.security.csrf.check.disabled.user.agents=OT DCC
```

4. Update the Keystore with Documentum Connector for Core (DCC) and Core configuration details:
   a. Open the command shell in admin mode.
   b. Navigate to the directory where the client configuration web application has been deployed, and then to the `d2keystore` subdirectory. For example, on Windows, this might be `C:\Tomcat\webapps\D2-Config\utils\d2keystore`.
   c. Run the following command to read the current Keystore properties and write them to a `d2keystore.properties` file.

      Note: The credentials you provide to the utility must be those of a superuser account in the global registry repository.

      ```
      D2KeyStoreUtil.cmd -u superusername -p password -r
      ```

      This command would generate `d2keystore.properties` in the same directory.
   d. Edit the file `d2keystore.properties` and add values for the following properties:
      - `dcc.user` - DCC admin user name
      - `dcc.password` - DCC admin password
      - `core.clientId` - OAuth2 client id registered with Core for current Smart View deployment

- `core.clientSecret` - OAuth2 client secret for current Smart View deployment

e. Run the following command to read the values from `d2keystore.properties` and write them to the Keystore:

```
D2KeyStoreUtil.cmd -u superusername -p password -w
```

f. Restart your app server.

5. Install `ExternalShare.dar` in the docbase. This .dar file is included in the Documentum Connector for Core build.

6. Update the `D2FS.properties` file in the location `D2-Smartview\WEB-INF\classes` with the Core URL and the DCC URL. Note that changes to the DCC URL will require an application server restart to take effect:

```
# Auth URL for Core OAuth2 integration.
# Please do not specify any value. It would refer to https://sso.core.opentext.com/
otdsws/login
coreOAuth2AuthUrl=

# Access token URL for Core OAuth2 integration.
# Please do not specify any value.  It would refer to https://sso.core.opentext.com/
otdsws/oauth2/token
coreOAuth2AccessTokenUrl=

# DCC Url for Shared to Core. Replace the ‘ipaddress’ and ‘port’ with DCC ip
address and port.
dccUrl= http://ipaddress:port/syncnshare-manual/v1
```

7. Update the `rest-api-runtime.properties` file in `D2-Smartview\WEB-INF\classes` with the following parameters:

```
#Must have the same value as given in d2fs-rest-web/web-inf/classes/rest-api-
runtime.properties
rest.security.crypto.key.salt=
rest.security.csrf.enabled=false
```

8. Make sure D2SV and DCC machines have access to internet so that it can access OpenText Core Share which is on the cloud. If these machines are proxy then follow these steps. Update `Catalina.properties` with ProxyHost, ProxyPort and nonProxyHosts.

Note: Configuration should be done in the `Catalina.properties` file in both the client application server on which Smart View is running and also on the DCC application server where the *syncagent* application is running.

`catalina.properties` in DCC App server - the nonProxyHosts point to the ContentServer host and the client application server host:

```
http.nonProxyHosts=<Documentum_Server_IP>|<AppServerIP>
https.nonProxyHosts=<Documentum_Server_IP>|<AppServerIP>
http.proxyHost=
http.proxyPort=
https.proxyHost=
https.proxyPort=
```

`catalina.properties` in App server running Smart View - the nonProxyHosts point to the ContentServer host and the DCC host:

```
http.nonProxyHosts=<DCC_IP>|<Documentum_Server_IP>
https.nonProxyHosts=<DCC_IP>|<Documentum_Server_IP>
http.proxyHost=
http.proxyPort=
https.proxyHost=
https.proxyPort=
```

9. Access core.opentext.com with administrator credentials.
10. Click the **Security** tab.
11. Create `OAuth` client by providing a description and redirect URL pointing to `D2-Smartview` - `http://<HOST>:<PORT>/ContextRoot` - where `ContextRoot` could be `D2-Smartview`.
12. Save the Client ID and secret key. This will be used in `D2FS.properties` in `D2-Smartview`.

Chapter 20

Digital signature components

This chapter includes information about digital signature plug-ins.

Important

- Smart View must run in HTTPS mode to allow callback from OpenText™ Core Signature.
- OpenText Core Signature requires JMS version 21.4 or higher.

20.1 Set up proxy in Java Method Server (JMS)

The Java Method Server (Tomcat server) must have an internet connection to access content from digital signature providers. If the JMS requires a corporate proxy to access the internet, you must add the following lines to *catalina.properties* inside the Tomcat `conf` folder.

```
http.nonProxyHosts=<Documentum_Server_IP>
https.nonProxyHosts=<Documentum_Server_IP>
http.proxyHost=
http.proxyPort=
https.proxyHost=
https.proxyPort=
```

20.2 Set up proxy in the Smart View/Client REST API Tomcat Server

The client server (Tomcat) must have an internet connection to access content from digital signature providers. If the JMS requires a corporate proxy to access the internet, you must add the following lines to *catalina.properties* inside the Tomcat `conf` folder.

```
http.nonProxyHosts=<Documentum_Server_IP>
https.nonProxyHosts=<Documentum_Server_IP>
http.proxyHost=
http.proxyPort=
https.proxyHost=
https.proxyPort=
```

Chapter 21

Install Microsoft 365 editing and co-authoring

Microsoft 365 editing enables the editing of files directly in the browser while within OpenText Documentum Content Management (CM). Files do not need to be downloaded, they are checked out automatically when opened and back in when closed. In addition, multiple users can edit the same file simultaneously, seeing each other's changes in real-time.

This chapter provides an overview of the installation and configuration steps for Microsoft 365 editing and co-authoring for Smart View.

Notes

- Depending on your licensing and configuration, this option might not be available.

To install and configure Microsoft 365 editing and co-authoring:

1. Install and configure OpenText Documentum CM Online Editing Service (OES): Section 3 "Deploying OpenText Documentum CM Online Editing Service" in *OpenText Documentum Content Management for Microsoft 365 - Deployment and Administration Guide (EEMSODC-IGD)* and Section 4 "Deploying Notification Service" in *OpenText Documentum Content Management for Microsoft 365 - Deployment and Administration Guide (EEMSODC-IGD)*.
2. Add the following to Admin Console's `rest-api-runtime.properties` file:

   ```
   rest.security.headers.x_frame_options.disabled=false
   rest.security.headers.x_frame_options.policy=SAMEORIGIN
   ```

3. Log in to Admin Console and configure Microsoft 365 editing and co-authoring: Section 2.1 "Configure editing and co-authoring in Microsoft 365" in *OpenText Documentum Content Management - Microsoft Integrations Administration Help (EDCADC-H-AIN)*.

Chapter 22

OpenText Documentum CM Mobile installation

OpenText Documentum CM Mobile (Mobile) is a lightweight mobile app that links directly to a client repository allowing users to browse, access, search and approve content on any iOS or Android device, from anywhere.

The Mobile browser client allows mobile users to access a client repository from mobile browsers such as Safari and Chrome.

22.1 Deploy Mobile on the AppWorks Gateway server

Before deploying Mobile, you must install and configure the AppWorks Gateway. See Section 5.1 "Installing Apps, Services, Components and Connectors" in *OpenText AppWorks Gateway - Installation and Administration Guide (OTAG-IGD)* for installation and configuration information.

Once the AppWorks Gateway is configured, you can deploy the `DocumentumMobile-<version>.zip` file. For instructions, see Section 5.1 "Installing Apps, Services, Components and Connectors" in *OpenText AppWorks Gateway - Installation and Administration Guide (OTAG-IGD)*.

Note: Before deploying `DocumentumMobile-<version>.zip` on AppWorks Gateway, you must uninstall any `d2mobile-<version>.zip` previously deployed. App versions 25.2 or earlier are not compatible with `DocumentumMobile-<version>.zip` .

22.1.1 Add a partition to the OTAG access role

1. Open OTDS to the Admin URL and log in with administrator credentials.
2. Select **Access Roles** from the left pane, and click **Actions** for the created OTAG access role.
3. Click **View Access Role Details**.
4. Click **Add**, select the required partition, and then click **Add selected Items to Access Role**.
5. Click **Save**.

22.1.2 Deployment options

There are two ways to deploy the AppWorks based Mobile hybrid app for Android and iOS operating systems. Both approaches have advantages and disadvantages. You can review the following information and choose the approach that meets your requirements.

Note: It would be mandatory to use the Smart View .war for Foundation REST API and not Client REST API .war, because when user shares document links via email, the links can only be opened in browser if the Smart View .war is deployed.

Table 22-1: Approach 1: Use Mobile app settings for Smart View URL

Pros	Cons
Same instance of the Smart View .war can be used for desktop/Mobile browser access as well as Mobile (AppWorks) access.	Smart View app server URL has to be exposed on the internet. This involves either a VPN, opening your organization's firewall, or putting the app server in the DMZ.

Table 22-2: Approach 2: Use AppWorks Gateway proxy

Pros	Cons
The Smart View app server does not need to be opened to the internet (in case VPN is not being used). All traffic is routed via AppWorks Gateway.	To access Smart View on Mobile browsers, users must use AppWorks proxy URL. Typically, this URL is only available on the internet. A special network configuration might be required to allow intranet access. Another option is to deploy another instance of Smart View .war for internal access only. In this case, all traffic is routed through the AppWorks Gateway and so more resources might be required to handle the traffic.

22.1.2.1 Approach 1: Use Mobile app settings for Smart View URL

In this approach, the Mobile app connects directly to the client app server URL for Client REST API services and only connects to the AppWorks Gateway for authentication / MDM purposes.

To use this approach, open the Mobile app settings and map the **Documentum Smart View URL** property to the app server in this format: `http://<app_server_ip:port>/<smart_view_context_name>`

22.1.2.2 Approach 2: Use AppWorks Gateway proxy

In this approach, the Mobile app only connects to the back-end Client REST API services via an AppWorks proxy. To use this approach:

1. In the AppWorks Gateway, configure the Proxy Rule to map path /d2 to Smart View app server URL. Use the following settings:
   - Whitelist: (?i)^(d2)(/.*|$)
   - Under **URL Mappings**, add the following mapping rule:
     – (?i)^(d2)/?(.*|$) -> http(s)://<app_server_ip:port>/<smart_view_context_name>/$2

       For example: (?i)^(d2)/?(.*|$) -> http://hostname:8080/D2-Smartview/$2
   - Allow Gateway to Handle OAuth 2.0: Unchecked
2. In the Moible app settings, keep the default value for Smart View URL - /d2.
3. In `WEB-INF/classes/settings.properties` file of the Smart View .war, add the following entry: `connection.remote.url = http://<AW_gateway_ip:port>/d2`

22.2 Two-factor authentication

Two-factor authentication (2FA) can be enabled for the Mobile client using OTDS configuration settings. OTDS has an embedded two-factor authentication solution that uses a time-based one-time password (TOTP) client application to obtain the authentication codes required to log in to the Mobile client. Two-factor authentication can also be implemented with a third-party two-factor authentication provider instead of OTDS' embedded solution. For more information about implementing two-factor authenticating, see the *OpenText Directory Services - Installation and Administration Guide (OTDS-IWC)*.

22.3 Cookie storage setting

For version 23.2 and later, to use the Mobile app, you must update how cookies are stored by browsers in the Smart View deployment.

To do so, set the *`samesite`* property to `none` in the `rest-api-runtime.properties` file. For example:

```
1 rest.security.client.token.cookie.samesite=none
```

Notes

- This change is only required if you use the Mobile app. Mobile accessed on a mobile browser does not require this update.
- Setting the *`samesite`* property to `none` does not impact client security.

- After you set the *samesite* property to `none`, users must access Smart View using HTTPS. If users access the Smart View client using HTTP, they will not be able to log in. This is a browser security restriction.

22.4 Install language packs

To install a language pack:

1. From the download center, download `DocumentumMobile-<version>.zip` and the `D2_Language_Packs_<language>_<version>.zip` for each of your required languages.
2. Extract `DocumentumMobile-<version>.zip`, then extract `mobile.zip`, which contains four subfolders: `csui`, `d2`, `esoc` and `mobile`. Note the location of your extracted files.
3. Extract `D2_Language_Packs_<language>_<version>.zip` and open the `ui` folder.
4. Copy the `csui`, `d2`, `esoc` and `mobile` folders from the `ui` folder into the extracted `mobile.zip` location from step 2.
5. Navigate to the `csui`, `d2`, `esoc` and `mobile` folders and check the `bundles/nls/<language>` folder in each, where you can see the translated string files.
6. Select all of the files from your `mobile.zip` extracted location and create a new `mobile.zip` file.
7. In the `DocumentumMobile-<version>.zip`, replace the existing `mobile.zip` file with your new `mobile.zip` file.
8. Upload your updated `DocumentumMobile-<version>.zip` file to the AppWorks Gateway. The Mobile language should change to the required one from settings.

22.5 Client installation

Before using Mobile on a device, the following must be configured:

- AppWorks Gateway
- Smart View configured with OTDS (required for login).

To install Mobile on a device:

1. Search for OpenText Documentum CM Mobile in the device's application store, such as the Play Store or App Store, and download the app.
2. Once installed, launch Mobile. When launched for the first time, the **Settings** page appears.
3. Enter the AppWorks Gateway Server URL in the placeholder for server address and tap the check mark. Once a connection to the AppWorks Gateway is established, you are redirected to the OTDS login page.
4. Enter the credentials and tap Sign In.

5. If authentication is successful, you are prompted to set an offline pin. An offline pin is used to login to the app when the device is offline.

   Note: For offline access in Mobile, ensure you select **Permit offline access on device** under **MDM Settings** in AppWorks Gateway.

6. Enter a valid six-digit pin and tap the check mark.

Chapter 23

Uninstall

1. Use Documentum Administrator or DQL/API to look for and remove the following artifacts:
   - All object types added by the client DAR, which are prefixed with `D2` and `D2_`
   - All methods added by the DAR, which are prefixed with `D2` and `D2_`
   - All jobs added by the DAR, which are prefixed with `D2` and `D2_`
   - Run the following DQL queries to make sure nothing exists:
     - `select * from dm_type where name like '%d2%'`

     Note: In the usual course of operation, it is no longer a requirement to delete preferences (such as d2c and/or x3) during upgrades or uninstall operations. Certain unusual scenarios might require this mitigation, so the option to delete preferences is still available.

     If you have installed the plug-ins, search for types related to PDF Configuration (c2) and Transfer Configuration (o2).
   - Once you removed all the mentioned artifacts, run the consistency checker job and check for errors.
2. On Documentum CM Server, refer to the installation log file to delete the .jar files that were copied to the Java Method Server.
3. On the web application server:
   a. Disable or uninstall the client and client configuration web applications if your web application server provides a user interface for removing the web application.
   b. Delete the folders that contained the client and client configuration files.

Chapter 24

Troubleshoot the installation

24.1 Collect installer debug logs

Once the installation is completed, installer debug logs are generated and saved in the same location as the `Documentum-Client-Installer` launch application in the `Logs` sub-directory. The log file, named `documentum-client-install-log`, will be generated in the user's home directory, for example, `C:/Users/Administrator/documentum-client-install.log`.

24.2 Remove old versions of ctx.cab from client configuration machines

Problem

Occasionally, old versions of the files in `ctx.cab`, an Active-X add-on used on the machines running client configuration are present and need to be manually deleted and re-installed.

Resolution

Locate the ctx add-on on the client machine where client configuration is accessed and delete them. The add-on can usually be found at `C:\Windows\Downloaded Program Files`. If they are not visible in Windows Explorer, delete them through command prompt. After deletion, access client configuration on the machine again to get a new version of ctx.cab.

24.3 Unable to access the client using Microsoft Internet Explorer

Problem

The client cannot run ActiveX controls on Microsoft Internet Explorer.

Cause

Microsoft Internet Explorer blocks the client URLs from running ActiveX controls and MSXML.

Resolution

Make the client URLs a part of the intranet or Trusted Security Zone to allow ActiveX controls and MSXML.

24.4 Files corrupting during export

Problem

When exporting a file from the repository to your local file system (any version), the file is corrupted. This issue exists in all compatible web servers except Tomcat 5.5.

Cause

While using the **Save As** dialog box, the session times out, and the file is corrupted.

Resolution

Configure the HTTP 1.1 connector `connectionTimeout` global setting for your web application server to wait longer before disconnecting the session.

While the parameter defaults to 60 seconds when not set, installation of the web server sets the parameter to 20 seconds. The documentation for your web server contains the default value and further instructions.

For example, in Tomcat 6.x:

1. Navigate to `<Tomcat installation path>/conf/` and open `server.xml`.
2. Locate the line `<Connector port="<port>" protocol="HTTP:/1.1" connectionTimeout="<timeout duration>" />`.
3. Change <*timeout duration*> to the duration you want in milliseconds, such as 60000.

24.5 Caching and file-cleaning services fail to operate

Problem

Caching services and temporary file-cleaning services fail to operate normally due to file deadlock.

Cause

If the client is deployed on multiple JVMs on the same application server or machine, the JVMs by default share the same folder and lock files from each other.

On a Linux environment, the error is caused frequently by JVMs being run by different users.

On Microsoft Windows systems, the error is caused by critical files being overwritten.

Resolution

Set up private Java temporary directories for each JVM instance.

To define a specific Java temporary directory, add the parameter `-Djava.io.tmpdir=/tmp/my_jvm_tmpdir` to the JVM launch command line.

24.6 Content transfer does not go through the Branch Office Caching Services server

Problem

Content transfer does not go through the Branch Office Caching Services server.

Cause

If the Accelerated Content Services server is not running, the Branch Office Caching Services server is not called and the client uses the application server servlet.

Resolution

To identify the issue, review the log, located by default in `C:/logs/d2.log` for the following line:

```
[DFC_ACS_NO_ACS_FOR_DOCBASE] Cannot find ACS servers for docbaseId=xxx
docbaseName=xxxxxxx from docbrokers
```

If the line appears and the Accelerated Content Services server is running, wait roughly 5 minutes for the client to re-try communication with the Accelerated Content Services server and then reconnect.

Running a Java Console and logging at level 5 may indicate why the connection is failing.

24.7 Slow file transfer when using a Linux-based operating system

Problem

File transfer is slower than Documentum Administrator when run on a Linux-based operating system.

Cause

Known issue with random number generation on Linux-based operating systems.

Resolution

1. Manually start the random generator daemon by typing `/sbin/rngd -b -r /dev/urandom -o /dev/random` as the root user.

   You can also include `-Djava.security.egd=file:///dev/./urandom` in your web application server startup script. Refer to your web application server documentation for further information.

2. If your Documentum CM Server is Linux-based, you may need to modify the Java Method Server to use the random number generator. Check the number of `entropy_avail` events by typing `cat /proc/sys/kernel/random/entropy_avail`

   If the `entropy_avail` did not increase, navigate to and open `startMethodServer.sh`, then add the line `Djava.security.egd=file:///dev/urandom`

   For Java 5 or later, use the line `-Djava.security.egd=file:///dev/./urandom`

24.8 No JMS server available exception when trying Java Method Server (JMS) failover

Problem

On a Documentum CM Server cluster environment, the client returns a `No JMS Server Available` exception when trying to perform an import operation. The JMS log shows an authentication error.

Cause

Authentication error for HA-JMS (High Availability JMS).

Resolution

1. Log in to Documentum Administrator.
2. Navigate to **Client Rights Management** > **Privileged Clients**.
3. Under **Manage Clients**, select **Enable trusted login** and **Enable trusted server privilege** for the Documentum CM Server Foundation Java API.

24.9 InstallException when installing the same DAR files

Problem

Installation of the same DAR files for the second time throwing error as: "InstallException: Unexpected error installing dar."

Cause

This happens when an existing project `D2-DAR` found in the workspace that is project `D2-DAR` already has core project **DocumentumCoreProject** in its classpath.

Resolution

Use the validated workaround for this problem:

1. Delete all the folders in `\Documentum\product\7.0\install\composer\ComposerHeadless\darinstallerworkspaces`.

2. Close the opened **dardeployer**.
3. Launch the **dardeployer** and try to install the same DAR again.

24.10 Server communication failure is occurring while saving dictionary or client URLs in client configuration

Problem

Saving or importing a data dictionary or saving client URLs in client configuration causes an error. The error might be similar to the following:

```
  1 com.documentum.fc.common.DfException: RECONNECT not possible for ticketed login.
m_reconnectNeeded=true thread=...
```

Cause

This problem can occur because the `login_ticket_timeout` setting is too low.

Resolution

Go to repository IAPI and increase the `login_ticket_timeout` by using the following command:

```
retrieve,c,dm_server_config
dump,c,l
set,c,l,login_ticket_timeout
60
save,c,l
dump,c,l
```

Clear the cache and restart the app server.

24.11 Smart View fails to load

Problem

Smart View fails to load, with the following error reported in the Smart View or application server log:

```
ERROR com.documentum.fc.client.security.impl.IdentityManager [] -
[DFC_SECURITY_IDENTITY_CREATION] failure on creation of identity: 'bad bundle'
com.documentum.fc.common.DfException: [DFC_SECURITY_IDENTITY_BUNDLE_FAIL] could not
create identity bundle because identity initialization failed
```

Cause

This Foundation Java API error is caused by an invalid or corrupted Keystore file.

Resolution

Take a backup of the `dfc.keystore` file present under D2-Smartview > WEB-INF > classes and then delete the file. If the Keystore file is shared among applications, back up the common `dfc.keystore` file and then delete it. Restart Smart View to create a new `dfc.keystore` file. Try loading Smart View again.

Chapter 25

Configuration files

Navigate to *<install path of web application server>*/webapps/D2-Config/WEB-INF/classes for the client configuration configuration files:

- D2-Config.properties
- dfc.properties
- logback.xml
- shiro.ini

Navigate to *<install path of web application server>*/webapps/D2/WEB-INF/classes for the client configuration and D2FS configuration files:

- applicationContext.xml
- settings.properties
- logback.xml
- D2FS.properties
- dfc.properties
- shiro.ini

  If you did not rename shiro-base.ini during configuration of authentication, shiro.ini may not exist.

Chapter 26

Appendix

This appendix provides reference information for:

- `D2FS.properties` settings reference
- `Settings.properties` settings reference
- Upgrade guidance for version 4.<*x*>

26.1 D2FS.properties settings reference

Most changes to `D2FS.properties` will autoload without application server restart. To see the effect of modified or updated settings, users must logout and then log back in. Changes can take up to 1 minute to take effect.

The following settings require an application server restart:

```
1. blockViewerRenditionRequest
2. c2ProcessingMaxPageThreshold
3. featureSecureProcessing
4. enableClientTimeZoneAwareness
5. localeFallback
6. showD2TasksOnly
7. includeWorkflowConfigName
8. cancelCheckoutVdConversionToSimpleDoc
9. searchTimeout
10. pretestJMSForMethods
11. DQLSearchAllowedQueries
12. extendDQLSearchAllowedQueries
13. hideDomain and hideDomain.YourRepoName1 , hideDomain.YourRepoName2
14. LoadOnStartup
15. dccURL
16. fileCleanSchedulerInterval
```

Parameter	Description	Applies to:
allow_concurrent_login	Setting this value to `false` will restrict users to one active session. If a user attempts to log in to more than one session, the second attempt will fail. By default, this parameter is set to `true`, meaning that concurrent logging in to different sessions is permitted.	Classic View

Parameter	Description	Applies to:
allowDetailedError	By setting this property to `false`, Client REST API prevents sending details in the error response. Default value is `false` (By default Client REST API will not send details in error response)	Smart View
allowedFileEndings	Allow list of file endings that are the only endings allowed. An empty list blocks all files. For example: `allowedFileEndings=docx,xlsx,txt,pdf,msg` **Note:** When copying or pasting existing content in the repository, the block list and allow list rules do not apply. Content already in the repository is not scrutinized..	Smart View, Classic View, Mobile

Parameter	Description	Applies to:
allowFilesWithNoExtension	Specify whether files can be imported without a file extension. This setting is ignored if both `allowedFileEndings` and `disallowedFileEndings` are not defined. If one of these is defined and `allowedFilesWithNoExtension` is not defined, a value of `false` will be understood. For example: `allowFilesWithNoExtension=false` This setting is `false` by default.	Smart View, Classic View, Mobile
allowRenditionRequest	Control multiple rendition requests for the same document. The default is `false`, which disallows multiple requests. `signoff` directs the client to check if the document is already signed off by the rendition server, then allows the request. If the sign off is not in place, multiple requests are denied. `true` allows the client to create a new rendition request for the same document without restriction.	Smart View, Classic View

Parameter	Description	Applies to:
application.tmp.dir	By default, the temp files location is created in the respective web application folder (`tomcat/temp/<webapp_name>`). If you want to use a different temp file location, set an absolute path appropriate for the deployment based on Windows or non-windows folder structure and convention (no shortcuts accepted). For example: `application.tmp.dir=C:\\TestTmp\\temp` **Note:** Any changes to this property requires application server restart.	Classic View
aspectsRequiringAppServerForUpload	Lists the aspect names that require upload through the app server.	Smart View, Classic View
blockViewerRenditionRequest	Controls the queue status of the rendition in `dmi_queue_item` from the PDF viewer. Set this parameter to `true` to block the rendition. Default is `false`. Any change to this setting requires an application server restart.	Smart View, Classic View

Parameter	Description	Applies to:
bravacsrViewerAllowChunking	Flag to indicate whether page chunking should be allowed or not See "Configure the BravaCSR viewer to "chunk" PDFs for viewer optimization (optional)" on page 112 for more information.	Smart View
bravacsrViewerChangemarkConfig	Location of the BravaCSR viewer changemark config file in the repository. Default is: `/System/BravaCSRViewer/config/ChangemarkConfig.xml`	Smart View, Classic View
bravacsrViewerChunkingChunksize	Number of pages to download in one chunk.	Smart View
bravacsrViewerChunkingTriggerFilesizeMB	Even when chunking is allowed, it is triggered only when either file size is more than `filesizeMB`, or total number of pages in the file is more than `pageCount`.	Smart View
bravacsrViewerChunkingTriggerPagecount	Even when chunking is allowed, it is triggered only when either file size is more than `filesizeMB`, or total number of pages in the file is more than `pageCount`.	Smart View

Parameter	Description	Applies to:
bravacsrViewerImageDirectory	Folder in the repository where images for BravaCSR viewer image annotation are stored. If this property is commented out, no server-side images will be fetched. Default is: `/System/BravaCSRViewer/Images`	Smart View, Classic View
bravacsrViewerImageFormats	File formats to be filtered for inside the `bravacsrViewerImageDirectory`. If this property is commented out, the list of included formats defaults to `bmp,gif,jpeg,png`.	Smart View, Classic View
bulkImportTimeToLive	When the temporary file clean up process runs, it deletes any files on its **remove** list that are older than this value (in minutes) if they were created as part of a folder structure bulk import. This value overrides the value set in *`timeToLive`*. The default value is `15` and the minimum is `5`. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Smart View, Classic View
cacheDocumentDql	Type a DQL query to find content to simulate location computation. For example: `cacheDocumentDql = dm_document where r_content_size > 102400 order by r_content_size asc`	Smart View, Classic View

Parameter	Description	Applies to:
cacheLocations	Type a list of cache locations separated by a comma. By default, this parameter uses the local IP address. For example: `cacheLocations = network1,network2`	Smart View, Classic View
cacheRepositoryList	Uncomment this property to cache the repository list from the DocBroker. The list can be refreshed from client configuration or by restarting the app server.	Smart View, Classic View
cancelCheckoutVdConversionToSimpleDoc	Converts a virtual document to a simple document on the `cancelCheckout` action using context menu option. If specified, and set to `true`, a virtual document is converted to a simple document on the `cancelCheckout` action. If not specified, or if set to `false`, a virtual document is not converted to a simple document on the `cancelCheckout` action. Any change to this setting requires an application server restart.	Smart View, Classic View
checkForMissingDocInWorkflow	Checks if a document in a workflow is deleted if the value is set to `true`. If set to `false`, this check is skipped.	Smart View, Classic View

Parameter	Description	Applies to:
clientProductName	Provide a product name to be saved in the audit table and sent to Documentum CM Server upon user login.	Smart View, Classic View
compressedExtensions	Type a list of file extensions separated by a comma. For example: `compressedExtensions=doc,docx,xls,xlsx,ppt,pptx,pdf,txt`	Smart View, Classic View
contentStagingLocation	Specify the shared folder location to store in-process files. If not specified, then the app server will store such files under `java.io.tmpdir`. If specified, it is assumed that the specified location references a mounted or shared directory that will be shared by all app servers in a cluster.	Smart View
contentTransferUrlTicketTimeout	Type a value to set the time in minutes that a download URL remains valid.	Smart View, Classic View
copyPasteWithAutolink	Allow or restrict autolinking during copy and paste operation. By default, this is set to `false`, disabling autolink during copy and paste of content. To apply autolink during copy and paste of content, uncomment this property and set to `true`.	Smart View, Classic View

Parameter	Description	Applies to:
coreOAuth2AccessTokenUrl	Access token URL for OpenText Core Share OAuth2 integration. Please do not specify any value. It would refer to `https://sso.core.opentext.com/otdsws/oauth2/token`	Smart View
coreOAuth2AuthUrl	Auth URL for OpenText Core Share OAuth2 integration. Please do not specify any value. It would refer to `https://sso.core.opentext.com/otdsws/login`	Smart View

Parameter	Description	Applies to:
coreOAuth2UIwithinIFrame	Controls how the login page for OpenText Core Share appears in the Smart View client. If the value is `true` (which is the default), the Core Share login page is shown inside an IFrame inside the dialog. If the value is set to `false`, a simple dialog with a message that a new window was opened will appear, and the user completes Core Share login in the new window. If a pop up blocker is enabled, the dialog will show an appropriate message. **Note:** If this value is set to `false`, you must ensure that the client URL does not fall under **Internet Zone** in the Internet Explorer browser, or the pop up behavior will be incorrect, and the Core Share login will not work. In such cases, the URL should be added under **Trusted Sites** in Internet Explorer.	Smart View
d2WebAppURL	URL that can be used to link back to objects when the client is installed behind a load balancer. For example, when links to object locations are exported to an Excel spreadsheet.	Classic View

Parameter	Description	Applies to:
dccUrl	DCC Url for Share to OpenText Core Share. Replace the `ipaddress` and `port` with DCC ip address and port, in this example: `http://ipaddress:port/syncnshare-manual/v1` **Note:** Any change to this setting requires an application server restart.	Smart View
dfc.application_code	Set to `dfc.application_code=dmc_rps` to allow users with retention policy services (RPS) privileges to add RPS objects in Admin Console.	Admin Console

Parameter	Description	Applies to:
dfcQueryCacheConsistencyCheckValue	Indicates how often the DQL query used in the **Properties** page needs to be checked for consistency with the server. If the value is numeric, it indicates the time (in seconds) that the caller is willing to allow the query to be used without being re-checked for consistency. If the requested query is found in the cache and has not been checked for consistency within the specified time period, it is checked and re-issued from the server, if necessary. If the value is non-numeric, (`check_never`, `check_first_access`, `check_once_per_scope`), it specifies a cache configuration object that defines the consistency check rules. There is a similar property in the `d2-jms.properties` file that also needs to be set with the same value for the refresh of the Foundation Java API query cache in order for it to be effective. **Note:** Setting this property risks slow **Properties** page performance.	Classic View

Parameter	Description	Applies to:
disableCleanupProcess	Enables or disables the temporary file clean up process. You can customize the process using the `.json` storage file. Set to `true` to disable clean up process and `false` to enable it. The default value is `false`. **Note:** Disabling this process does not disable the tracker task that updates the temporary files to the `.json` storage file for every task. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Smart View, Classic View

Parameter	Description	Applies to:
disallowedFileEndings	Block list of file endings that are not allowed: an empty list allows all files. This setting is ignored if the `allowedFileEndings` setting is defined. **Note:** When copying or pasting existing content in the repository, the block list and allow list rules do not apply. Content already in the repository is not scrutinized. The following files are disallowed by default: `bat, com, exe, js, hta, html, htm, jar, vbs, vb, sfx, dll, tmp, py, msi, gadget, cmd, vbe, jse, ps1, ps2, ps1xml, ps2xml, psc1, psc2, lnk, inf, scf`	Smart View, Classic View, Mobile
displayCreationProfileForSelectedFolder	Specify the direction to list the display of creation profiles. If set to false, the display of creation profiles will be from the parent cabinet until the selected folder (i.e., Top-Down). If set to true, the display of creation profiles will be from the selected folder until its parent cabinet (i.e., Bottom-Up).	Classic View

Parameter	Description	Applies to:
docbase.connection.monitor	This property enables the client to monitor the Foundation Java API client connections threshold (`dfc.session.max_count`). If the active connection count exceeds the threshold percentage, it starts releasing the connections that are not active in the transaction so that the system does not result in a No More Session API. **Note:** This property works only if you enable `dfc.connection.profiling=true` in `dfc.properties`. To apply the property, uncomment it and set it to `true`.	Classic View
docbase.connection.threshold	This property indicates how much threshold of active connection is allowed. It should be given as a percentage. Once the active connection count reaches the threshold, the client cleans up the connections.	Classic View

Parameter	Description	Applies to:
DQLSearchAllowedQueries	DQLs added to selectDql and locateDql parameters of intelligent URLs will be considered only if the property `extendDQLSearchAllowedQueries` is set to `true` and if the dql is part of this property's value which must be a bar separated list of desired DQLs. For example: `select r_object_id from dm_document where object_name='Test_Document.docx'| select r_object_id from dm_user where object_name='Test3AA.pdf'` **Note:** The allow list DQL statements do not accept the use of `%` as wildcard for pattern matching in the like clause. For example, the following would be an invalid allow list DQL entry since it uses `%` in the like clause as a wildcard for pattern matching: `select r_object_id from dm_document where object_name like '%mall%'` Any change to this setting requires an application server restart.	Classic View

Parameter	Description	Applies to:
dumpGroupNames	Users in the listed groups are allowed to perform an information dump in the Smart View client. Multiple groups can be configured, comma separated. No group listed means all users are allowed to perform a dump if `enableDump=true`. For example: `dumpGroupNames=group1,group2`	Smart View
EnableBubbleMenuForTasks	Enables or disables the display of Classic View's hover menu in the Task list widget. Default is `true` (menu is enabled).	Classic View

Parameter	Description	Applies to:
enableClientTimeZoneAwareness	By default, the system tracks date and time based on the server's time zone. Enable this setting to display dates and times according to the user's local time zone. To enable, set to `true`, which is the default setting. When the setting is enabled, the system will convert the server's time to the user's local time zone when displaying dates and time. This ensures that users see date and time values based on their own time zone settings without altering the server's actual time. The setting affects all areas of the application where dates and times are displayed such as property pages. Any change to this setting requires an application server restart.	Smart View, Classic View
enableDump	Enable or disable the Smart View client information dump service. Default is `false`.	Smart View

Parameter	Description	Applies to:
extendDQLSearchAllowedQueries	DQLs added to `selectDql` and `locateDql` parameters of intelligent URLs will be considered only if this property is set to `true` and if the dql is part of `DQLSearchAllowedQueries` property. For example: `http://host:port/D2?docbase=docbaseName1&login=userName&password=password&selectDql=select r_object_id from dm_document where object_name='Test_Document.docx'` If this property is set to `false`, any DQLs added to URL will get rejected even if it is present in `DQLSearchAllowedQueries`. Any change to this setting requires an application server restart.	Classic View

Parameter	Description	Applies to:
enableTimeBasedCleanup	Enable or disable the temporary file clean up process time-based initiation. If enabled, the process runs based on the value set in *fileCleanSchedulerInterval* whether or not the temporary files exceed the thresholds set in either *maxTempFiles* or *maxTempSizeMB*. Once initiated, the temporary file clean up process deletes all files on the **remove** list that exceed the value defined in *timeToLive*. The default value is `false`. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Smart View, Classic View
fileCleanSchedulerInterval	Used for background file cleaner thread. Default is 15 minutes, so that the background thread runs in 15 minute intervals. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Classic View

Parameter	Description	Applies to:
hideDomain	Set to `true` to hide the domain for login. You can also specify the repository by using the parameter `hideDomain.<RepositoryName>`. Any change to this setting requires an application server restart.	Smart View, Classic View
hideMonitoringDialogForAdmin	Hides the displayed **Monitoring** dialog. The option overrides all the configurations on the **Monitoring** menu item and hides it when set to true.	Smart View, Classic View
ignoreSearchOrderBy	Set the property to true to ignore search Order By. If the property is unset, the default value is false.	Smart View, Classic View
includeSearchPreferenceColnames	For search by qualification using IURL, if the search should only include default column values and ignore the user's search column preference, set the property to false. The property is set to true by default, which will include the user's previously searched column data along with the default columns.	Classic View

Parameter	Description	Applies to:
includeWorkflowConfigName	The following columns are included in the tasklist soap response using `getFilteredContentResponse` when this flag is set to true: `workflow_id`, `workflow_name`, `workflow_config_name`. Any change to this setting requires an application server restart.	Classic View
inputSantiyTags	Controls attribute input values to apply sanitization. If the input value contains any tags, it is applicable for the sanitization process and sanitization is performed based on the `antisamy.xml` configuration policy file. The property values are separated by a comma.	Smart View, Classic View
ios_locate_url	Configures the send mail message body to include iPhone specific locate_url. `$value(ios_locate_url)` will resolve to iPhone url and append the `'\<r_object_id>` Eg `$value(ios_locate_url)` will resolve to `x-otm-as-d2mobile://?launchUrl=nodes/09005bc1800ab505`. For example: `ios_locate_url=x-otm-as-d2mobile://?launchUrl=nodes`	Classic View

Parameter	Description	Applies to:
irmProtectedFormats	Sets Information Rights Management for selected file types, including: `pdf`, `msw8`, `excel18book`, `ppt8`, `msw12`.	Smart View, Classic View
irmTicketTimeout	This property sets the IRM login ticket time out. The default values is 5 minutes.	Smart View, Classic View
isSMTPRequired	By default, imported .msg files with generated mail rendition (.eml) recipients display in exchange address format and display the exchange address in the `From, To, CC & Bcc` sections. For example: `User Name </O=OPENTEXT/OU=NORTH AMERICA/CN=RECIPIENTS/CN=USERNA1>` Set the flag to `true` to view the recipient's email address in SMTP format, `User Name <userna1@opentext.com>`.	Classic View
justInTimePDFChunkingValidation	For existing documents with missing metadata to support chunking, a profile request queue is set up just in time when the document is accessed in `bravaCSRViewer`.	Smart View
limitDelegateAsynchronousToUsers	Loads the Delegate to Users list asynchronously. The default value is true. To disable the property, set the value to false.	Classic View

Parameter	Description	Applies to:
loadBalancedContentServer	Flag that identifies your Documentum CM Server environment as being load balanced. Default is `false` (your environment is not load balanced). Set flag to `true` if you have employed Documentum CM Server load balancing. **Note:** If you have set up a high availability multi-Documentum CM Server environment, ensure that this flag is set to `true` to avoid `[DM_SESSION_E_AUTH_FAIL]` errors from appearing when users view PDFs in the PDF Viewer widget.	Smart View, Classic View
loadDelegateToUsersAsynchronously	Loads the Delegate to Users list asynchronously.	Smart View, Classic View

Parameter	Description	Applies to:
LoadOnStartup	Add the name of the repositories (comma separated) to initialize those repositories on startup. All variables for each repository need to be defined in the global registry keystore. For example: `LoadOnStartup=repository1,repository2` The following variables need to be defined in the global registry keystore: `LoadOnStartup.<repository>.username`, `LoadOnStartup.<repository>.password` and `LoadOnStartup.<repository>.domain` Any change to this setting requires an application server restart. **Note:** Workflow reporting features might not operate as expected for completed or aborted workflows if this setting is not enabled.	Smart View, Classic View

Parameter	Description	Applies to:
localeBasedDateFolderInAutoLink	Indicates whether locale should be used or not during date folder creation according to autolink configuration. This value is commented out and set to `true` by default. To have date folders in autolink use only the English locale and not create any other locale based date folders, uncomment this property and set to `false`. For example, if autolink configuration has YYYY of effective_date for folder and the need is to have only English digits for year as 2021 and not Arabic/Hebrew digits for a year , then this property should be uncommented and set as `false`. `localeBasedDateFolderInAutoLink=false` This property also appears in the `d2-jms.properties` file. It behaves the same as it does in the `D2FS.property` file. Enter the same value in both files.	Classic View

Parameter	Description	Applies to:
localeFallback	• If `localeFallback` language is specified and browser language is not installed on Documentum CM Server listed in `DD_LOCALES`, the client falls back to the specified language. • If not specified, the labels are used for the user locale regardless of `DD_LOCALES` and the config ID is returned when the localized configuration is missing in the existing behavior. • If `localeFallback` language is specified, setting should contain only one language code. For example: `#localeFallback=en` Any change to this setting requires an application server restart.	Smart View, Classic View
logExceptionForMissingRendition	This property should be set to false if the user does not wish to see the exception stacktrace for `D2MissingRenditionException`.	Smart View, Classic View

Parameter	Description	Applies to:
login.repository.default	Sets a default repository. For Classic View, this property takes precedence over the deprecated property of the same name in `settings.properties`, and if `login.repository.default` is also set in `settings.properties`, it will be ignored. For Smart View, setting `login.repository.default` in `D2FS.properties` will cause this repository to be selected by default in the repository selector on the sign in page.	Smart View, Classic View
maxBrowserWidgetResultSetSize	Sets the maximum result set size for the Browser widget when users are browsing into the tree : If unset, the default value is `maxResultSetSize`.	Smart View, Classic View
maxEntriesForPresetProfiles	Specify the maximum number of preset profile records to be shown in the UI. Default is `8`	Smart View
maxEntriesForRecentlyUsedProfiles	Specify the maximum number of recently used profile records to be shown in the UI. Default is `5`	Smart View

Parameter	Description	Applies to:
maxFolderDepthForCreationProfiles	Specify the maximum folder depth level from the cabinet level when displaying creation profiles. If commented out, a value of 0 is understood. Any integral value is valid, and a negative value means that the maximum folder depth level is effectively infinite. By default it is set to 4.	Smart View, Classic View
maxObjectsForMassUpdatePropertyFetch	Specify the number of objects the system can accept for mass update property page property expansion functionality. If the number of object id for the request is greater than the given property value then expansion will not work. Client REST API will not be sending a property value for each object ID. The default value is `1000`	Smart View
maxRecycleBinWidgetResultSetSize	Sets the maximum result set size for the recycle bin widget before filtering is used. If unset, the default value is `maxResultSetSize`.	Smart View, Classic View

Parameter	Description	Applies to:
maxResultSetSize	Type a value to limit the result set of all queries used in populating the Users and Groups widgets as well as user and group selection lists in property dialog boxes. Use this parameter to avoid performance problems associated with large result sets. The default value is `1000`. **Note:** This is applicable to similar widgets, such as repository browser widget.	Smart View, Classic View
maxTaskResultSetSize	Maximum result set size for the Tasks widget when viewing the user's tasks. If unset, default value == `maxResultSetSize`. **Note:** Limiting the task list size would have an impact on the task folder displayed in the Task browser.	Smart View, Classic View
maxTempFiles	Set the maximum number of temporary files that can be stored in the temporary location of the application server. The default value is 2000. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Smart View, Classic View

Parameter	Description	Applies to:
maxTempSizeMB	Set the threshold for maximum total file size (in MB) of temporary files. Once met, the temporary file clean up process runs and deletes all files on the **remove** list that exceed the value defined in *timeToLive*. The default value is 1024 and the minimum value is 10. This parameter is also included in the D2-Config.properties and D2-BOCS.properties files.	Smart View, Classic View
maxUploadFileSize	Do not allow users to upload a file larger than maxUploadFileSize (in bytes). The default is to have restrictions. For example: maxUploadFileSize=10485760.	Smart View, Classic View, Mobile

Parameter	Description	Applies to:
maxUploadRequestSize	Sets the maximum upload request size in bytes (default value is 16GB) This enables the prevention of denial of service attacks launched by uploading very large files. **Note:** File upload to the app server is implemented using a multipart form post, so the size of a file upload request body will be slightly larger than the file itself. This setting is used when the multipart form post request is parsed. If the value of the Content-Length request header exceeds the maximum upload request size, the request is rejected. If the value of this setting is zero or negative, no restriction is imposed on the size of a multipart form request.	Smart View, Classic View
maxUserSelectionPerPerformerType	Maximum User selection set size for Performer type while sending to the workflow. If unset, default value == -1	Smart View, Classic View

Parameter	Description	Applies to:
maxWorkflowResultSetSize	This setting controls the maximum result set size for the My Workflow and All Workflows list in the Smart View client only. If this is unset, the default value == `maxResultSetSize`.	Smart View
noConversionToVdWithVersionPermit	If specified and set to `true` a user with Version permit is not allowed to convert a simple doc to virtual document and gets a failure message of insufficient access. If not specified or set to `false` a user having Version permit on a simple document can convert it to a virtual document. On adding a grandchild_doc, the child_Doc is converted to a virtual document.	Smart View, Classic View

Parameter	Description	Applies to:
objectCreationLocation	Allows you to specify a fixed location for initial object creation. By default, Documentum CM Server implicitly links new objects to the user's defined home cabinet/folder. If users do not have WRITE access to their home cabinet/folder, this setting can be used to specify a temporary location where objects will be created. When the creation process is complete, the object will be re-linked to its final location as usual (for example, through auto-linking or other functionality). **Note:** This folder must be pre-existing and all users must have at least WRITE permission to the folder. It will not be created.	Smart View, Classic View
pluginsOrder	Type a list of plug-ins by name to force the order in which they are loaded. For example, if you have plugin1 that computes data during a property save and plugin2 that verifies data during a property save, you want verification to occur after computation. In this example, set the line as: `pluginsOrder=plugin1,plugin2`	Smart View, Classic View

Parameter	Description	Applies to:
printPendingRecords	Enables/disables printing all pending temporary files on every run. Prints on `LOG4j.properties` INFO mode. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Classic View
processXploreResultSet	Set this flag to `true` if post processing of the search result set is needed. Setting this to `true` is useful if there is a need to fetch checkout status of a document in search result set. Note that this may slightly degrade the performance of search. If unset, the default value is taken as `false`. **Note:** Applicable to xPlore search only.	Smart View, Classic View
propertyPageConstraintValidation	Default is `true` (enabled). Set to `false` to disable the validation constraint. This setting ensures that disabled, hidden, or read-only (immutable) properties cannot be saved if the initial value sent to the requesting system is not the same value returned.	Smart View, Classic View

Parameter	Description	Applies to:
searchColsDoclistConfigName	For search by qualification using IURL, if the search should include a specific doclist column's values and ignore the user's search column preference, set the property to the doclist's config name. This property will only be considered if `includeSearchPreferenceColnames` is set to false and the doclist config name is similar to the doclist widget name as per client configuration. By default, if no value is specified, the client will fall back to the default set of column names.	Classic View
searchTimeout	A maximum time to wait in milliseconds. If not set, default will be 0 and search will have no time out. If set, a `DfSearchException` error is thrown if the timeout elapses. Any change to this setting requires an application server restart.	Smart View, Classic View
showD2TasksOnly	Set this flag to `true` to display tasks triggered from the client. Setting the flag to `false` displays tasks triggered from other products. Any change to this setting requires an application server restart.	Smart View, Classic View

Parameter	Description	Applies to:
simpleSearchDql	Allows you to make search options configurable when **Enable full text search** is not enabled in client configuration's **Interface > Search** menu. To override the default search dql, specify `simpleSearchDql=<<`*`your_custom_dql`*`>>` using `$value(<`*`search_term`*`>)` to reference the search term.	Smart View, Classic View
taskSubjectRecomputation	Setting this to `false` will ignore the `$USER, $NOW ,$TODAY` and `user_preference` locale changes of the user. The default value is `true`. *For group task*: It shows the same subject to all the users on specific task. Subject computation would happen based on the first user who views the task (In case of the group task) . Creation of Task folders for respective users in the group tasks would be done only when the user loads the task browser *For user task*: A single subject is displayed for all the locales of the user.	Smart View, Classic View
templateFilterPropertyNames	Sets the Template Filter combo attributes. The default value is set to `object_name,title, subject`.	Classic View

Parameter	Description	Applies to:
timeToLive	When the temporary file clean up process runs, it deletes any files on its **remove** list older than this value (in minutes). The default and minimum value is `5`. This parameter is also included in the `D2-Config.properties` and `D2-BOCS.properties` files.	Smart View, Classic View
userProfileImageExtensions	Allows restrictions to Client REST API and the Smart View user profile image file types. Specify the applicable image formats or extension for user profile image for upload. For example: `userProfileImageExtensions=bmp,jpeg,jpg,png,gif,svg`	Smart View
useSQLServerPerfHint	Replaces "RETURN_TOP" with "SQL_DEF_RESULT_SET" hint in doclist DQL to improve SQLServer performance. Default is `false`. Set to `true` to turn on the hint, and add the following setting in server.ini: `"SQL_DEF_RESULT_SET_AND_OBJECT_BASED=1"`	Smart View, Classic View
validateDelegationCreation	Validates delegation creation on creation of a new OOO delegation to see if any delegation is already created for the given date range. To enable the property, uncomment it and set it to true.	Smart View, Classic View

Parameter	Description	Applies to:
validateDeletePermission	If `false`: **Delete All version** radio button will be enabled upon deletion of any object. If `true` - **Delete All version** radio button will be enabled only when all the current and all version of the selected document(s) has delete permission of the logged in user.	Classic View
Validator.format	Validates whether given value can be a `dm_format.name` value or `dm_sysobject.a_content_type` value This regular expression requires that the value have at least 1 character and at most 32 characters, where every character being a letter, digit, underscore, or hyphen. `dm_format.name` values can have up to 64 characters, but `dm_sysobject.a_content_type` values must have no more than 32 characters. If commented out, no validation will be performed, but this introduces a DQL injection security vulnerability.	Smart View, Classic View

Parameter	Description	Applies to:

workflow.planning.disabled	Set the flags to remove workflow planning for all workflows or a comma separated list of workflows. If the value is `true` then the planning option is removed for all workflows. If the value contains a list of workflows, the planning option will be removed for only the listed workflows. For example:
	
	`workflow.planning.disabled=true`
	
	`workflow.planning.disabled=(Legal Review WF),(WF HR Validation)`
		Classic View

workflowReportingTimeout	The user interface displays a popup message warning the user of a long running workflow search query. The default value is 60 seconds. This value must be specified in milliseconds.	Smart View
workflowWithTBO	Set to true, if TBO is attached to the workflows. **Note:** In case of sequential tasks, update performer operation is disabled for workflows with TBO.	Smart View, Classic View
x3GadgetServerUrl	Allows the override of the `x3_options.gadget_server_url` property in a repository on a per App Server basis.	Classic View

26.2 settings.properties settings reference

Changes to `settings.properties` will autoload without application server restart. To see the effect of modified or updated settings, users must logout and then log back in. Changes can take up to 1 minute to take effect.

Parameter	Description
browser.folder.limit	Type the limit for the number of folders displayed in any single level of the Repository browser widget. If the end user views a folder containing more items than the limit set, the client shows a **More** button.
browser.plugin.activex	Set to `true` to enable users to export content to the desktop using drag and drop feature. This setting enables dragging content to the desktop using an ActiveX plug-in. This feature is only supported in Internet Explorer 9 and later.
browser.plugin.mode	Set to one or more of the content transfer modes. See *Documentum Content Transfer Framework Supported Feature List* in the Administration Guide for more information. **Note:** If you do not set a mode, the system defaults to thin • `wsctf` • `thin` The indicated modes in `settings.properties` determines which clients the users will be able to choose from in the **D2 Client > User Settings**. In addition, the order will determine the fallback process the client will follow. For example, `browser.plugin.mode=wsctf,thin` would attempt wsctf first, then fall back to thin. See the `settings.properties` file for further details.
checkin.rlockmachine.check	Set rlockmachine validation when checking in files. Introduced for Mapped Network Drives. If `true`, rlockmachine validates when checking in files. If `false`, no rlockmachine. The check will be performed when checking in files using the Java plug-in. If the line is commented out, the effective value is `true`.

Parameter	Description
connection.remote.url	Uncomment and type the address of the proxy server to enable content transfer in a reverse proxy setup. `http<s>://<proxy or server address><:<port>>/<D2>`
csp.header.value	The value of this property is emitted as the value of the `Content-Security-Policy` HTTP response header when a user accesses the client with a browser. If this property is not specified (or is commented), the default value emitted as the value of the `Content-Security-Policy` response header is `frame-ancestors=none`. The properties `allowed.frame.origins` and `X-Frame-Options` used in earlier versions are now discontinued, as the `csp.header.value` setting affords the same results. The `settings.properties` file contains detailed information about the header.
download.folderexport.batchsize	Set the maximum number of document download URLs returned in the initial folder export request. This limit prevents the client from timing out the request when processing a very large number of documents. Default is `300` documents.
enableRichtextEditorSourceEdit	Enable or disable Source Edit mode for the RichText/HTML editor field in Properties pages. This is applicable for all the Properties pages. Valid inputs are `true` and `false`, and the default value is `false`. If set to true, the **Source Edit** mode switch is available and the user can add HTML source code and preview it.  **Caution** Enabling the mode allows users to enter malicious HTML script code. Previewing the code will execute the code and could potentially compromise security.
error.uncaught.display	Set to `true` to show uncaught error messages.

Parameter	Description
geolocation.enabled	Default value is `false`. Set to `true` to enable the geolocation feature where the user's location at the time of login is reported back to Documentum CM Server and is saved in the audit log. Enabling this setting without also enabling `geolocation.required` in `settings.properties` allows users to decline location tracking in their browser, but still log in. **Note:** Geolocation information for SSO and IURL logins is not supported. The geolocation API is restricted to https environments as per the browser security, so only secure origins are allowed for geolocation.
geolocation.required	Default value is `false`. This setting is only applicable if the `geolocation.enabled` setting in `settings.properties` is enabled. Set to `true` to require users to allow location access through their browser before they can log in. User failure to provide access restricts login and causes an explanatory error message to appear. **Note:** Geolocation information for SSO and IURL logins is not supported. The geolocation API is restricted to https environments as per the browser security, so only secure origins are allowed for geolocation.
import.prefer.sameproperties	Set to `true` if user is importing multiple email files with same profile and properties checkbox selected but needs email files to retain their individual attributes (For example, through the Transfer Configuration plug-in). **Note:** File attributes take precedence over inheritance and default value template.
language.user.forced	Append the two-letter language code if you want to force users to access the client in a specific language and disable language options.

Parameter	Description
login.domain.hide	Set to `true` to hide the domain for login. You can also specify the repository by using the parameter `hideDomain.<RepositoryName>`.
login.networklocation.hide	Set to `true` to show the network location selector. When administrator changes the network location settings in Documentum Administrator, the Accelerated Content Services settings cache is updated every two minutes by default. For testing convenience and to reflect the change immediately in the login dialog, you can add the following line to `dfc.properties`: `dfc.acs.gr.refresh_interval = 0` For production machine, we recommend to use the default refresh interval, that is two minutes.
login.repository.default	Type a repository name to set the default login repository.
login.repository.filter	Type the name of the repositories (separated by comma) that you want to hide from the login screen. For example, Global Registry repository. **Note:** If you add the default repository name in the list of hidden repositories, the default repository will be hidden from the login screen. That means, `login.repository.filter` parameter overrides the value of the `login.repository.default` parameter.
mobile.view.enabled	Enables the client for use in a mobile browser. By Default the parameter `mobile.view.enabled=true` (mobile view will be enabled by default) If `mobile.view.enabled=false` mobile view will be disabled If `mobile.view.enabled` is totally removed from `settings.properties`, then it will default to mobile view `=true` and will be enabled.

Parameter	Description
ondbclickEvaluateDocumentMenuView	This property is used for menu evaluation on double click operation performed on Doclist /DocGallery widget. Default value is `false`. Selected file will be downloaded on double click. Set to `true` allows users to evaluate the selected document to ensure that the view menu item is available in the `MenuDocument` or `MenuContext` types and handles download.
sessiontimeout.counter	The counter in seconds for the session timeout warning dialog in Smart View and Admin Console. Default is `60`. `sessiontimeout.counter = 60`
showRelationColumnInRelationAssistance	This setting is used to display the relation name column in Classic View's Relation Assistance dialog while launching a workflow. The default value is `false`. **Note:** The administrator should delete the preferences of the users and default preferences to opt-in to this feature.
transfer.http.compression	Set to `true` to enable HTTP compression.
uid.session.cookie.timeout	Set the time in seconds that the session remains valid after a user closes or refreshes a browser tab or navigates away in a browser tab. If the browser itself (no open tabs or windows) is closed the session is lost immediately.

26.3 BravaCSR viewer ChangemarkConfig.xml file sample

```
  1 <?xml version="1.0" encoding="utf-8"?>
  2 <ChangemarkConfiguration xmlns="http://www.infograph.com/xml_schemas/
Brava_ChangemarkConfiguration_1p0p0">
  3 <Types>
  4 <ChangemarkType name="Action">
  5 <State name="For Discussion" defaultState="true">
  6 <Color>
  7 <red>195</red>
  8 <green>235</green>
  9 <blue>255</blue>
 10 </Color>
 11 </State>
 12 <State name="An Idea">
 13 <Color>
 14 <red>95</red>
 15 <green>255</green>
```

```
<blue>150</blue>
</Color>
</State>
<State name="Investigate">
<Color>
<red>255</red>
<green>190</green>
<blue>110</blue>
</Color>
</State>
<State name="Typo">
<Color>
<red>255</red>
<green>205</green>
<blue>160</blue>
</Color>
</State>
<State name="Revision Error">
<Color>
<red>255</red>
<green>250</green>
<blue>180</blue>
</Color>
</State>
<State name="Confirm">
<Color>
<red>180</red>
<green>140</green>
<blue>200</blue>
</Color>
525
Configuring changemark markup
</State>
<State name="Urgent">
<Color>
<red>255</red>
<green>90</green>
<blue>80</blue>
</Color>
</State>
</ChangemarkType>
<ChangemarkType name="Change" >
<State name="Request" defaultState="true">
<Color>
<red>195</red>
<green>235</green>
<blue>255</blue>
</Color>
</State>
<State name="Issued">
<Color>
<red>255</red>
<green>255</green>
<blue>255</blue>
</Color>
</State>
<State name="Approved">
<Color>
<red>95</red>
<green>255</green>
<blue>150</blue>
</Color>
</State>
<State name="Rejected">
<Color>
<red>255</red>
<green>90</green>
<blue>80</blue>
</Color>
</State>
<State name="Superseded">
```

```
 87 <Color>
 88 <red>235</red>
 89 <green>210</green>
 90 <blue>130</blue>
 91 </Color>
 92 </State>
 93 <State name="Released">
 94 <Color>
 95 <red>0</red>
 96 <green>0</green>
 97 <blue>0</blue>
 98 </Color>
 99 </State>
100 <State name="Verified">
101 <Color>
102 526
103 Using the xCP Advanced Viewer
104 <red>180</red>
105 <green>140</green>
106 <blue>200</blue>
107 </Color>
108 </State>
109 <State name="Completed">
110 <Color>
111 <red>0</red>
112 <green>0</green>
113 <blue>0</blue>
114 </Color>
115 </State>
116 <State name="Closed">
117 <Color>
118 <red>100</red>
119 <green>200</green>
120 <blue>255</blue>
121 </Color>
122 </State>
123 <State name="Notice">
124 <Color>
125 <red>0</red>
126 <green>0</green>
127 <blue>0</blue>
128 </Color>
129 </State>
130 </ChangemarkType>
131 <ChangemarkType name="Agreement">
132 <State name="Acceptable" defaultState="true">
133 <Color>
134 <red>95</red>
135 <green>255</green>
136 <blue>150</blue>
137 </Color>
138 </State>
139 <State name="Can't Do">
140 <Color>
141 <red>0</red>
142 <green>0</green>
143 <blue>0</blue>
144 </Color>
145 </State>
146 <State name="Carve Out">
147 <Color>
148 <red>255</red>
149 <green>250</green>
150 <blue>180</blue>
151 </Color>
152 </State>
153 <State name="Typo">
154 <Color>
155 <red>255</red>
156 <green>90</green>
157 <blue>80</blue>
```

```
158 527
159 Configuring changemark markup
160 </Color>
161 </State>
162 <State name="Discuss Further">
163 <Color>
164 <red>195</red>
165 <green>235</green>
166 <blue>255</blue>
167 </Color>
168 </State>
169 <State name="Make Reciprocal">
170 <Color>
171 <red>195</red>
172 <green>235</green>
173 <blue>255</blue>
174 </Color>
175 </State>
176 <State name="Reword">
177 <Color>
178 <red>255</red>
179 <green>90</green>
180 <blue>80</blue>
181 </Color>
182 </State>
183 </ChangemarkType>
184 <ChangemarkType name="Issue">
185 <State name="Undetermined" defaultState="true">
186 <Color>
187 <red>195</red>
188 <green>235</green>
189 <blue>255</blue>
190 </Color>
191 </State>
192 <State name="Critical">
193 <Color>
194 <red>255</red>
195 <green>90</green>
196 <blue>80</blue>
197 </Color>
198 </State>
199 <State name="High Priority">
200 <Color>
201 <red>255</red>
202 <green>230</green>
203 <blue>95</blue>
204 </Color>
205 </State>
206 <State name="Med Priority">
207 <Color>
208 <red>255</red>
209 <green>240</green>
210 <blue>215</blue>
211 </Color>
212 </State>
213 <State name="Low Priority">
214 528
215 Using the xCP Advanced Viewer
216 <Color>
217 <red>195</red>
218 <green>235</green>
219 <blue>255</blue>
220 </Color>
221 </State>
222 <State name="Closed">
223 <Color>
224 <red>0</red>
225 <green>0</green>
226 <blue>0</blue>
227 </Color>
228 </State>
```

```
229 </ChangemarkType>
230 <ChangemarkType name="Status">
231 <State name="Investigate" defaultState="true">
232 <Color>
233 <red>195</red>
234 <green>235</green>
235 <blue>255</blue>
236 </Color>
237 </State>
238 <State name="Pending">
239 <Color>
240 <red>195</red>
241 <green>235</green>
242 <blue>255</blue>
243 </Color>
244 </State>
245 <State name="Working">
246 <Color>
247 <red>195</red>
248 <green>235</green>
249 <blue>255</blue>
250 </Color>
251 </State>
252 <State name="Review">
253 <Color>
254 <red>195</red>
255 <green>235</green>
256 <blue>255</blue>
257 </Color>
258 </State>
259 <State name="Completed">
260 <Color>
261 <red>0</red>
262 <green>0</green>
263 <blue>0</blue>
264 </Color>
265 </State>
266 <State name="Closed">
267 <Color>
268 <red>0</red>
269 <green>0</green>
270 529
271 Configuring changemark markup
272 <blue>0</blue>
273 </Color>
274 </State>
275 </ChangemarkType>
276 <ChangemarkType name="Missing">
277 <State name="Signature">
278 <Color>
279 <red>255</red>
280 <green>90</green>
281 <blue>80</blue>
282 </Color>
283 </State>
284 <State name="Account No">
285 <Color>
286 <red>255</red>
287 <green>90</green>
288 <blue>80</blue>
289 </Color>
290 </State>
291 <State name="Verification">
292 <Color>
293 <red>255</red>
294 <green>90</green>
295 <blue>80</blue>
296 </Color>
297 </State>
298 <State name="Amount">
299 <Color>
```

```
300 <red>255</red>
301 <green>90</green>
302 <blue>80</blue>
303 </Color>
304 </State>
305 <State name="Approval" defaultState="true">
306 <Color>
307 <red>255</red>
308 <green>90</green>
309 <blue>80</blue>
310 </Color>
311 </State>
312 <State name="Address">
313 <Color>
314 <red>255</red>
315 <green>90</green>
316 <blue>80</blue>
317 </Color>
318 </State>
319 <State name="Other">
320 <Color>
321 <red>255</red>
322 <green>90</green>
323 <blue>80</blue>
324 </Color>
325 </State>
326 530
327 Using the xCP Advanced Viewer
328 </ChangemarkType>
329 <ChangemarkType name="Project">
330 <State name="Deliverable" >
331 <Color>
332 <red>195</red>
333 <green>235</green>
334 <blue>255</blue>
335 </Color>
336 </State>
337 <State name="Cost/Benefit">
338 <Color>
339 <red>195</red>
340 <green>235</green>
341 <blue>255</blue>
342 </Color>
343 </State>
344 <State name="Feasibility">
345 <Color>
346 <red>195</red>
347 <green>235</green>
348 <blue>255</blue>
349 </Color>
350 </State>
351 <State name="Concern" defaultState="True">
352 <Color>
353 <red>195</red>
354 <green>235</green>
355 <blue>255</blue>
356 </Color>
357 </State>
358 <State name="Resources">
359 <Color>
360 <red>195</red>
361 <green>235</green>
362 <blue>255</blue>
363 </Color>
364 </State>
365 <State name="Estimate">
366 <Color>
367 <red>195</red>
368 <green>235</green>
369 <blue>255</blue>
370 </Color>
```

```
371 </State>
372 <State name="Scheduling">
373 <Color>
374 <red>195</red>
375 <green>235</green>
376 <blue>255</blue>
377 </Color>
378 </State>
379 </ChangemarkType>
380 </Types>
381 </ChangemarkConfiguration>
```

26.4 Upgrade guidance for version 4.x

Refer to the following chart if you are upgrading from a 4.x version of the software. Read from top to bottom.

Note: All the versions must be first upgraded to version 4.5 before running the Migration utility and before upgrading to version 21.4.

Table 26-1: Upgrade guidance for version 4.x

Upgrade tasks	4.0	4.1	4.2	4.5	4.6	4.7
Upgrade to version 4.2	Req [1]	N/A	N/A	N/A	N/A	N/A
Upgrade Documentum CM Server to supported version	N/A	DS 7.0, 7.1 [4]	DS 7.0, 7.1, 7.2 [5]	DS 7.0, 7.1, 7.2, 7.3	DS 7.1, 7.2, 7.3	DS 7.1 and higher
Upgrade to version 4.5	[2]	Req	Req	N/A	N/A	N/A
Run the Migration utility [3]	Req	Req	Req	Req	N/A	N/A
Upgrade to the latest version	Req	Req	Req	Req	Req	Req

Note: [1] Version 4.0 is not supported on a 7.x platform. As such, version 4.0 must first be upgraded to a version that is supported on the currently underlying the Documentum CM Server platform. If the current Documentum CM Server platform is pre 6.7 SP2, upgrade to version 4.2, else upgrade to version 4.5.

[2] If you're not yet at version 4.5, this step is required.

[3] Migration is an irreversible process. Restore to pre-migration state is possible only through database backup and restore, or any other snapshot type restore. The Migration utility does not impact files on the file system, it only affects client configuration objects in the repository. To run the utility, follow the instructions in "Run the Migration utility" on page 24.

[4] Documentum CM Server 7.1 is supported with version 4.1 P18 and higher.

[5] Documentum CM Server 7.2 is supported with version 4.2 P17 and higher.