

OpenText™ Documentum™ Content Management

Records Client Deployment Guide

Set up the environment, deploy, and configure the Records Client applications as a unified solution.

EDCRM250400-IGD-EN-01

OpenText™ Documentum™ Content Management Records Client Deployment Guide

EDCRM250400-IGD-EN-01

Rev.: 2025-Oct-16

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>

© 2025 Open Text

Patents may cover this product, see <https://www.opentext.com/patents>.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Table of Contents

1	Overview	7
1.1	Intended audience	7
2	Quick start	9
3	Planning for deployment	11
3.1	Required and optional supporting software	11
3.2	Typical configuration	11
3.3	Installing JDK	12
3.3.1	Using SSL communication	12
3.4	Preparing Documentum CM Server	13
3.5	Application server host requirements	13
3.5.1	Directory name restriction	13
3.5.2	Content transfer directory permissions	13
3.5.3	DNS resolution	14
3.6	Configuring Documentum Secret Integration Service	14
3.7	Configuring OpenText Directory Services	16
3.8	Deploying multiple applications	16
3.9	Deploying language packs	17
3.9.1	Installing the DAR with localized data dictionary properties files	17
3.9.2	Populating and editing locales in data dictionary	17
3.9.3	Applying language packs	17
3.10	Customizing an application	18
4	Preparing the client hosts	21
4.1	Ensuring a certified JVM on browser clients	21
4.2	Supporting Outlook mail message	22
4.3	Using Citrix Presentation Server Client	22
4.4	Using browser extension-based content transfer mechanism	22
4.5	Installing browser extension and native client application	22
5	Preparing the application server host	23
5.1	Setting the Java memory allocation	23
5.2	Turning off failover	23
5.3	Preparing environment variables for non-default Foundation Java API locations	24
5.4	Preparing JBoss	24
5.4.1	Configuring Red Hat JBoss Enterprise Application Platform (JBoss EAP)	24
5.4.2	Classloader settings	26
5.4.3	Supporting JDK with JBoss EAP and Elytron security update	27
5.5	Preparing WildFly	29

5.5.1	Configuring WildFly on Windows	29
5.5.2	Classloader settings	31
5.6	Preparing Tomcat	32
5.6.1	Configuring Tomcat for JDK	32
5.6.2	Disabling tag reuse	32
5.6.3	Disabling HttpOnly property	33
5.6.4	Disabling whitespace parsing	33
5.7	Preparing IBM WebSphere Liberty	33
5.7.1	Configuring for the Jakarta-supported version of IBM Websphere Liberty	33
5.8	Preparing to use an external web server	34
6	Installing the Records Client products	35
6.1	Overview	35
6.2	Primary steps	36
6.2.1	Download the Records Client files	36
6.2.2	Creating the global repository	41
6.2.3	Creating the Records Client repository	41
6.2.4	Installing the DAR files with the DAR Deployer plug-in	42
6.2.5	Installing the DAR files using Headless Composer on Linux	45
6.2.6	Configuring Records Client to connect to the Vault server	45
6.2.7	Installing records.war on the application server	46
6.2.8	Importing WebFormService.jar into Documentum CM Server	47
6.2.9	Setting up Documentum Administrator	48
6.2.10	Registering all Foundation Java API clients	48
6.3	Configuring users, roles, and audits	48
6.4	Troubleshooting the new installation	49
6.5	Upgrading from earlier releases	50
6.5.1	Upgrade paths and overview	50
6.5.2	Upgrading and migrating previous versions of the Records Client to the latest versions	50
6.5.3	Migrating custom record relationship instances	51
6.6	Configuring security	53
6.7	Working with the new email processing functionality	53
6.7.1	Working with new email messages imported from Records 20.4 onwards	53
6.7.2	Configuring the MailApp.properties file	53
6.7.3	Using configuration options in app.xml for email import	55
6.8	Configuring Records Activator for declaring email records from Microsoft Outlook	56
6.9	Configuring Records Manager for barcode label printing and batch processing	58
6.9.1	Configuring the NiceLabel environment variable	58

6.9.2	Setting up barcode label printing (rendering) and testing instructions ..	59
6.9.2.1	Installing Adobe Print Driver and NiceLabel	59
6.9.2.2	Testing barcode label printing	60
6.9.2.3	Optional steps to import custom NiceLabel Templates	62
6.10	Batch processing XML schema for Records Manager	63
6.10.1	Overview	63
6.10.2	Batch processing operations	69
6.10.2.1	Overview of operation status and view report	69
6.10.2.2	Reconciliation	74
6.10.2.2.1	Resolvable issues	76
6.10.2.3	Charge-in	76
6.10.2.4	Mark as destroyed	78
6.10.2.5	Mark as exported	79
6.10.2.6	Mark as shipped	80
6.10.2.7	Move to container	81
6.11	Updating the pass-along workflow	82
6.12	Configuring the disposition job to run successfully against objects with an Unknown disposition strategy	83
6.13	Configuring the parameters for the Retention Policy Services notification job	83
6.14	Setting up Records Queue Manager	83
6.14.1	Installing and configuring Records Queue Manager in silent mode	88
6.14.2	Unconfiguring and uninstalling Records Queue Manager services	91
6.14.3	Unconfiguring and uninstalling Records Queue Manager services in silent mode	92
7	Upgrading a WDK-based application	95
8	Deploying a WDK-based application	97
8.1	Preparing the WAR file for deployment	97
8.2	Enabling Foundation Java API connections to repositories	98
8.3	Enabling Foundation Java API memory optimization	100
8.4	Configuring UCF	100
8.5	Enabling presets and preferences repositories	101
8.5.1	Configuring encrypted password for presets and preferences repositories	102
8.6	Enabling retention of folder structure and objects on export	103
8.7	Working with email messages in WDK-based applications	104
8.8	Enabling external searches	104
8.8.1	Configuring the connection to the search server	104
8.8.2	Configuring the connection to the backup search server	105
9	Completing the deployment	107
9.1	Deploying default virtual link support	107

9.2	Accessing the application	108
9.3	Testing WDK samples	108
9.4	Configuring user authentication	109
9.5	Licensing OpenText Documentum CM	109
10	Installing Application Connectors	111
10.1	Overview	111
10.2	GUI installation of Application Connectors	112
10.3	Commandline installation of Application Connectors	114
11	Installing Documentum Webtop Extended Search	115
11.1	Installing the Webtop Extended Search option	116
11.2	Viewing installation log files	118
12	Enabling the Webtop Express DAR	119
13	Troubleshooting deployment	121
13.1	Wrong JRE used for application server	121
13.2	No global registry or connection broker	121
13.3	No connection to repository	122
13.4	Login page incorrectly displayed	122
13.5	Slow performance	122
13.6	Out of memory errors in console or log	123
13.7	Slow display first time	123
13.8	Foundation Java API using the wrong directories on the application server	123
13.9	Tag pooling problem	123
13.10	UCF client problems	124
13.11	Citrix client problems	124
13.12	Connection issues between an ECI Services server and IPv6 clients	124
13.13	Presets not working	125
14	Deploying a custom application	127
14.1	Using the comment stripper utility	127
15	Installing Brava	129
A	Predeployment checklist	131

Chapter 1

Overview

Records Client is the unified client and a web application built on WDK and can therefore be customized using WDK. The deployment process is largely the same as for other WDK-based applications. For more information on developing or customizing applications that are WDK-based, refer to the latest *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*. “Installing the Records Client products” on page 35 describes how to deploy the Records Client for the GA release.

Instructions are included for deploying `records.war`, installing records DAR files, upgrading, and configuring the records products. All other chapters are available for general reference regarding the planning of the deployment, preparing servers, and so on.

For more information, see the product *Release Notes*, available at OpenText My Support (<https://support.opentext.com>), for system requirements and notes that may affect deployment.

1.1 Intended audience

This guide is intended for administrators and includes instructions for setting up the Records Client, enabling a repository for use, setting up label printing and batch processing, and setting up Records activator for Microsoft Outlook. To deploy a WDK-based application, you should be familiar with the operating system of the application server and be able to install and configure a J2EE application server.

Chapter 2

Quick start

This chapter describes the steps you need to perform to deploy your application. The steps are described in more detail in the chapters of this guide. Your product or environment may require additional steps, which you can find in the product-specific chapter or chapters of this guide or in the index. To install the DAR files of the records products on the Records Client, refer to [“Installing the Records Client products” on page 35](#). It does not matter which operation is completed first as long as the DAR files are installed and the WAR file is deployed.

To perform a simple product deployment

1. Plan the deployment. (Refer to [“Planning for deployment” on page 11](#).)
Check that you have required and optional supporting software, prepare the OpenText™ Documentum™ Content Management Server, check application server environment requirements, prepare for multiple applications, plan for language pack deployment, and (if supported) plan to deploy a customized application.
2. Prepare the clients. (Refer to [“Preparing the client hosts” on page 21](#).)
Install a supported browser virtual machine and perform specific browser preparations for Microsoft® Edge and Mozilla® Firefox®. If needed, you will install the mail message converter and prepare Citrix® clients.
3. Prepare the application server. (Refer to [“Preparing the application server host” on page 23](#).)
Ensure you have sufficient memory allocated to the application server Java instance, turn off failover if it is not needed, and follow application-server and proxy-server specific preparation instructions.
4. Deploy the product WAR file using the application server standard deployment mechanism. (Refer to [“Deploying a WDK-based application” on page 97](#).)
You must first unpack the WAR file archive and enter some information that is specific to your environment: your connection broker and global registry information, optional presets and preferences repositories, and optional ECIS search server.
5. Optionally, if your product has a WAR file and DAR files, make sure to deploy the product DAR files using the DAR Deployer.
6. Complete the deployment. (Refer to [“Completing the deployment” on page 107](#).)
After successful deployment, you can configure UCF, deploy root virtual link support, and test the application samples.

Chapter 3

Planning for deployment

Provides information about software and hardware requirements to deploy a WDK-based application. Also, contains instructions that are shared by all WDK-based products. Check your release notes for information on the application servers, browsers and other software in the environment that are certified for your product.

3.1 Required and optional supporting software

Prerequisites before deploying WDK and WDK applications:

- OpenText Documentum Content Management (CM) Server and its associated database
- Documentum CM Server global repository
- Connection broker

You must specify one or more connection brokers in the `dfc.properties` file. Refer to [“To configure connections in `dfc.properties` before deployment:” on page 99](#) for information on configuring the connection broker before deployment.

- J2EE application server or servlet container
- Oracle JDK/OpenJDK. For more information, see [“Installing JDK” on page 12](#).

All WDK-based applications require DARs that must be installed in the repository.

3.2 Typical configuration

When deployed on a single application server, a typical WDK-based application requires the following network components:

- Application server host on which the WDK-based application will be deployed
- Separate Documentum CM Server host, where a repository is installed and where one or more Documentum CM Servers run
- Global registry repository
- Client hosts that run a supported web browser

[Figure 3-1](#) shows the network components.

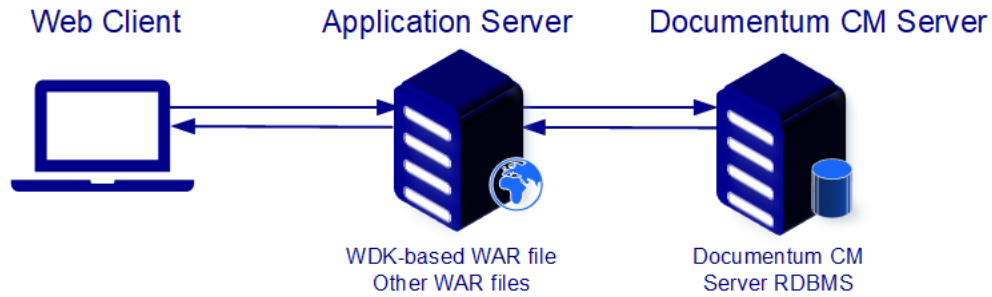


Figure 3-1: Basic WDK host configuration



Caution

For security and performance reasons, it is advised to install the Documentum CM Server and a WDK-based application on separate hosts. In addition, the Documentum CM Server installs an internal JBoss server, which cannot be used to deploy web applications for licensing reasons.

Clustered environments

WDK-based applications can be deployed in supported clustered environments. For more information about the WDK-based application and supported managed server configurations, see the product *Release Notes*.

3.3 Installing JDK

Download and install the supported version of JDK from Oracle JDK/OpenJDK website. For more information, see *Oracle JDK and OpenJDK documentation*.

3.3.1 Using SSL communication

OpenText recommends to use certificate-based SSL communication. *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)* contains more information about certificate-based SSL communication.

For SSL communication, by default, OpenText™ Documentum™ Content Management Foundation Java API searches the value of `dfc.security.ssl.truststore` (property for certificate) in `dfc.properties`. If found, OpenText Documentum Content Management (CM) Foundation Java API uses the certificate-based SSL communication. Otherwise, Foundation Java API defaults to the cipher-based anonymous SSL supported by Java.

3.4 Preparing Documentum CM Server

Documentum CM Server installs certain DARs that are required for a WDK-based application. You do not need to perform a separate installation of these. Products built on WDK or Webtop may require additional DARs, which are available on the My Support.

Global registry requirement

A global registry must be installed in your environment in order to run a WDK-based application. A global registry is a Documentum CM Server that has been designated as a global registry. For information on designating your application's global registry before deployment, see [“Enabling Foundation Java API connections to repositories” on page 98](#).

3.5 Application server host requirements

The application server host used for WDK-based applications has the requirements described in the following sections.

3.5.1 Directory name restriction

Java does not allow directories containing the following characters, which must not appear in the directory names or paths of OpenText Documentum CM applications:

```
! \ / : * ? " < > |
```

3.5.2 Content transfer directory permissions

The content transfer directory on the application server host is used to store files temporarily when they are transferred between the repository and the client machine. The default content transfer directory is specified in the `app.xml` file as the value of `<server>.<contentlocation>`. The application server instance owner must have write permissions on this temporary content transfer location.

You can change the default value to a location on your application server host for which the application server instance owner has write permission. For information on specifying locations in the UCF client and server configuration files, see *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*.

Some application servers require policies that grant permissions to write to these directories. Refer to deployment information for your application server to see OpenText Documentum CM policy settings.

3.5.3 DNS resolution

The Domain Name Server (DNS) must be configured to properly resolve IP addresses based on the web address used to access the server.

3.6 Configuring Documentum Secret Integration Service

Documentum Secret Integration Service (DSIS) is a standalone program that acts as a connector between OpenText Documentum CM products and HashiCorp Vault. The DSIS file is packaged with the Documentum CM Server installer. DSIS must be running on all machines that needs it.

To configure Documentum Secret Integration Service:

1. Download the Documentum CM Server installer from My Support.
2. Extract the installer to a temporary location.
3. Copy the `dsis.zip` file from where you extracted the installer to a temporary location.
4. Extract the `dsis.zip` file to a folder named `dsis`.
5. Configure the DSIS daemon agent.

To update the parameters in the `<dsis>/application.properties` file to configure the DSIS daemon agent and to authenticate with Vault, do the following:

- a. `dsis.dctm.host`: Host of the DSIS daemon agent. DSIS daemon agent always runs on the localhost only. Make sure that the value is set to `localhost`. Do not change this value.
- b. `dsis.dctm.port`: Port used for the DSIS daemon agent connection. By default, the value is set to 8200. However, you can change to any available port. Make sure that the port you change is the same port used for the DSIS daemon agent connection.
- c. `dsis.dctm.executorThreadCount`: Executor thread count used for the load requirement. By default, the value is set to 10. Do not change this value.
- d. `dsis.dctm.token`: DSIS authentication token. To generate the DSIS authentication token, go to the `dsis` folder and run the following command:

Windows

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

Linux

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

Copy the generated token and provide it as a value for this parameter.

- e. `dsis.dctm.kvpath`: Obtain the Key-Value (KV) path from the Vault server and provide it as a value for this parameter.

For more information about Key-Value path in Vault, see *HashiCorp Vault* documentation.



Note: You can configure individual secrets to have its own Key-Value path.

- f. Retain the default values for the following parameters:

- `dsis.dctm.tokenNeeded`
- `dsis.dctm.retryFailure`
- `dsis.dctm.retrySleepInterval`
- `dsis.dctm.enforceListDuringInit`

- g. Update the authentication method values in the `spring.cloud.vault` parameters. OpenText supports role-based, token-based, Kubernetes token, and TLS authentication methods.

For more information about the authentication method parameters, see *Spring Cloud Vault* documentation.

6. Start the DSIS daemon agent using the following command:

Windows

```
dsis_start.bat >> dsis.log
```

Linux

```
dsis_start.sh >> dsis.log
```

7. Access the DSIS daemon agent using the following URLs including the HTTP header, *dsis-daemon-token*, with the value generated in [step 5.d](#):

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

The response code 200 for the preceding URL indicates that the DSIS daemon agent is initialized.

- `http://localhost:8200/dsis/secret/<secret_name>/<key_name>`

The preceding URL provides the password, if the secrets information with the key is found. For example, if you use the `http://localhost:8200/dsis/secret/DOCBASE_PASSWORD/docbase1` URL, it returns the repository password. For more information about secret and key names that must be stored in Vault, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)*.

8. Tracing information of DSIS is captured in the `<dsis>/dsis.log` file. To enable debugging of the Spring library, add the following entries in the application.properties file:

```
logging.level.ROOT=${LOG_LEVEL_ROOT:warn}
logging.level.org.springframework=${LOG_LEVEL_SPRING:debug}
logging.level.org.springframework.web=${LOG_LEVEL_SPRING_WEB:debug}
logging.level.org.springframework.security=${LOG_LEVEL_SPRING_SECURITY:debug}
```

Vault-enabled Records will use the following passwords from Documentum CM Server:

- CLIENTS_PREFERENCE_PASSWORD
- CLIENTS_PRESET_PASSWORD
- DFC_GLOBALREGISTRY_PASSWORD
- INSTALL_OWNER_PASSWORD

3.7 Configuring OpenText Directory Services

OpenText Directory Services (OTDS) is an authentication server that supports standard OAuth 2.0. The OAuth 2.0 authorization framework is the industry-standard protocol for authorization. It enables third-party application to obtain limited access to an HTTP service.

For the steps to configure OTDS, see the *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)*.

3.8 Deploying multiple applications

You can deploy multiple WDK-based applications on a single host. Each instance of an application must be deployed to a different virtual directory. If the applications share the same application server instance, the applications must be the same version or later.

You can deploy applications to separate instances of the application server. If the applications use different versions of Foundation Java API, you must deploy them in separate application server instances.

3.9 Deploying language packs

Instructions to deploy the Records Client and records products it hosts are available in [“Installing the Records Client products” on page 35](#).

Language packs are available to localize WDK-based applications. A language pack is a language-specific archive file that contains a graphical user interface (GUI) and user documentation that have been localized into a language other than the default application language, U.S. English.

To deploy language packs, unpack your product WAR file and add the language packs according to the specified instructions.

3.9.1 Installing the DAR with localized data dictionary properties files

To install the DAR files that have DAR Language Pack:

1. Download the core DAR file for the required product from My Support.
2. Download the corresponding DAR Language Pack from the same location as the product Language Pack.
3. Extract the DAR language pack ZIP file to a temporary directory to get the locales folder, which contains the localized properties files.
4. Install the DARs for the Records Client components in the following order: OpenText™ Documentum™ Content Management Retention Policy Services and OpenText™ Documentum™ Content Management Records Manager.
5. Browse and select the *Locales Folder* parameter in the DAR installer and provide the absolute file path to the localized properties files.

3.9.2 Populating and editing locales in data dictionary

You can populate, publish, and edit the locales in the data dictionary. For more information, see *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)*.

3.9.3 Applying language packs

To prepare the WAR file for deployment, see Documentum Web Development Kit and Webtop Deployment Guide.

To apply language packs:

1. Download the language pack ZIP file from My Support to a temporary directory on the application server host.
2. Extract the language pack to the temporary location where the WDK application WAR file is unpacked.

3. Go to <temporary_location>/WDK.
4. In a text editor, open the app.xml file and add the required locale under <supported_locales>.

For Example: If you have extracted German Language Pack, add German locale as follows:

```
<supported_locales>
  <locale>en_US</locale>
  <locale>de_DE</locale>
</supported_locales>
```

The supported locales are:

- French: fr_FR
- Italian: it_IT
- German: de_DE
- Spanish: es_ES
- Japanese: ja_JP
- Korean: ko_KR
- Simplified Chinese: zh_CN
- Brazilian Portuguese: pt_BR
- Arabic: ar_SA



Note: The value of the rightToLeft attribute must be set to true to enable the RTL mode for a locale. For example, the RTL mode for Arabic locale must be set as follows:

```
<locale rightToLeft="true">ar_SA</locale>
```

5. Repackage the WAR file. For more information, see *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*.

3.10 Customizing an application

A developer license is required to develop custom applications. Contact OpenText Global Technical Services to obtain a developer license.

Configuration

Configuration is defined for support purposes such as changing an XML file or modifying a Java Server Page (JSP) to configure controls on the page.



Note: Configuration does not require a developer license.

Customization

Customization is defined for support purposes as the extension of WDK classes or the modification of JSP pages to include new functionality. Customization requires a developer license.



Note: Customization of Documentum Administrator is not supported.

Chapter 4

Preparing the client hosts

This chapter contains instructions that are shared by all WDK-based products. For more information on the browsers that are certified for your product, see *Release Notes*, available on My Support (<https://support.opentext.com>).

4.1 Ensuring a certified JVM on browser clients

Browser client hosts require a certified version of the Java virtual machine (JVM or VM) to initiate content transfer in a WDK application. New machines may not have a JVM installed in the browser. For more information about your product version for the JVMs that are supported, see *Release Notes*, available on My Support (<https://support.opentext.com>).

If the WDK-based application is configured to use UCF content transfer, a lightweight applet is downloaded to the browser when the client makes the first content transfer or preferences request.



Note: UCF has a dependency on Microsoft Visual C and C++ runtime libraries. WDK-based applications like OpenText Documentum Content Management (CM) Records Manager that use UCF require the Microsoft Visual C++ Redistributable as a prerequisite. Microsoft has consolidated the Microsoft Visual C++ Redistributable packages for versions 2015 to 2022 into a single package. Download this package from the Microsoft website for the appropriate architecture (x86 or x64).

On Windows clients, if the JVM required for UCF is not present on the client machine, UCF uploads to a Windows client a private JVM. This VM does not replace the JVM that is used by the browser. For non-Windows browser hosts with a JVM of 17.x and later, you must pre-install a supported version of the Oracle JRE that will be used by UCF.

Since the UCF VM file (JRE) is over 10 MB in size, the installation can cause a delay. You can avoid this delay by installing a compatible local JVM prior to using UCF transfer.

4.2 Supporting Outlook mail message

See *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* for more information about the email processing functionality. For performing any outbound operations on the email messages (such as, view, export, drag and drop), the client host must have Microsoft Outlook client installed on it.

4.3 Using Citrix Presentation Server Client

Citrix Presentation Server Client can be used as a web browser. For more information about your WDK-based product to determine whether Citrix clients are supported for your product, see *Release Notes*, available on My Support (<https://support.opentext.com>).

In the Citrix environment, content files are exported or checked out to the Presentation Server host, not to individual client hosts. Each individual user works on a client host with an image of a web browser that is running on the Presentation Server host. For more information on enabling applications on Presentation Server, see to documentation provided by Citrix.



Note: If you have previously attempted to content transfer to the client, it will use the client's location machine, and you must delete the ucf directory file path that was installed on the local client machine under the OS home directory of the user, for example, C:\Documents and Settings\<username>\Documentum\ucf.

4.4 Using browser extension-based content transfer mechanism

See the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* for more details on browser extension-based content transfer mechanism.

4.5 Installing browser extension and native client application

See the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* for more details on installing browser extension and native client application.

Chapter 5

Preparing the application server host

This chapter describes the tasks you must complete to prepare the application server host before deploying your WDK-based application, and contains instructions that are shared by all WDK-based products. The product *Release Notes*, available on My Support (<https://support.opentext.com>), provides more information on the application servers that are certified for your product.

Before you deploy a WDK-based application, ensure that your J2EE application server or servlet container is a supported version and that it can successfully serve sample JSP pages.

Your selected application server and optional external web server must be certified for Records Client 25.4.

OpenText does not provide support for installing or running application servers. Contact the application server vendor for technical support.

5.1 Setting the Java memory allocation

OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD) provides more details to set the Java memory.

5.2 Turning off failover

If your application server and environment combination does not support failover, you can turn off failover in app.xml. Refer to the product *Release Notes*, available on My Support (<https://support.opentext.com>), to determine whether failover is supported for your environment.

If you do not turn off failover, you may see failover validation messages in the application server log, but these should not interfere with operations.



Note: Do not attempt to use the application in a failover environment that is not certified.

To turn off failover for the application, open app.xml in the custom directory and add the following element:

```
<failover>
  <enabled>false</enabled>
</failover>
```

5.3 Preparing environment variables for non-default Foundation Java API locations

The base location for content transfer on the application server host is specified by the Foundation Java API environment variable `dfc.data.dir`. This location is specified as the value of the key `dfc.data.dir` in `dfc.properties` located within the application WAR file in `WEB-INF/classes`. If this variable is not set in the environment for the application server, the default location is the OpenText Documentum CM subdirectory of the current working directory. (The current working directory contains the application server executable.) For example, in Tomcat the location is `%CATALINA_HOME%/bin`.

By default, the checkout and export directories are subdirectories of the `dfc.data.dir` directory, and the user directory is the same as `dfc.data.dir`. If you need to use non-default locations to change the default location, you can create environment variables for `dfc.checkout.dir`, `dfc.export.dir`, and `dfc.user.dir`, respectively. The default value of `dfc.registry.mode`, which corresponds to the key `dfc.registry.mode` in `dfc.properties`, is `file`. The full path to this file by default is `dfc.user.dir/documentum.ini`. For a non-default file name or location, specify it as the value of the environment variable `dfc.registry.file`.

5.4 Preparing JBoss

The product *Release Notes* contains information on the supported versions of JBoss, Tomcat, and tc Server.

5.4.1 Configuring Red Hat JBoss Enterprise Application Platform (JBoss EAP)

1. Download the JBoss EAP 8 Update 4 patch file named `jboss-eap-8.0.4-runtime-maven-repository.zip` from the Red Hat website.
2. Extract the `jboss-eap-8.0.4-runtime-maven-repository.zip` file to a temporary location.
3. Apply the JBoss EAP 8 Update 4 patch.

Windows

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

```
jboss-eap-installation-manager.bat update perform ^
--dir C:\jboss-eap-8.0 ^
--repositories mrrc::file:C:\jboss-eap-8.0.4.GA-runtime-maven-repository
\maven-repository ^
--offline
```

Linux

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

```
./jboss-eap-installation-manager.bat update perform \
> --dir ../../jboss-eap-8.0 \
> --repositories mrrc::file:C:\jboss-eap-8.0.4.GA-runtime-maven-repository
\maven-repository \
> --offline
```

4. If available, delete the `dfc.keystore` and `wdk.keystore` files in `<JBoss EAP_Home>\bin` on Windows or `<JBoss EAP_Home>/bin` on Linux.
5. Before creating the `web-inf-classes.jar` file, update the `dfc.properties` file.

Windows

Create an encrypted password using the following command:

```
java -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.base/java.lang=ALL-UNNAMED -cp .;../lib/dfc.jar;../lib/commons-io-<packaged-version>.jar; ../lib/commons-lang-<packaged-version>.jar; ../lib/bcprov-jdk<packaged-version>.jar TrustedAuthenticatorTool <PASSWORD>
```

where *<packaged-version>* is the supported version packaged with the Records Client WAR file.

Linux

Create an encrypted password using the following command:

```
java -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.base/java.lang=ALL-UNNAMED -cp ../lib/dfc.jar:../lib/commons-io-<packaged-version>.jar: ../lib/commons-lang-<packaged-version>.jar: ../lib/bcprov-jdk<packaged-version>.jar TrustedAuthenticatorTool <PASSWORD>
```

where *<packaged-version>* is the supported version packaged with the Records Client WAR file.

If Vault is enabled, the presets and preferences passwords are retrieved from Vault. If Vault is not enabled, you must manually update the encrypted password in the `wdk/app.xml` file for the presets and preferences passwords.

6. Move the keystore files from `<WebApp_Root>\WEB-INF\classes` on Windows or `<WebApp_Root>/WEB-INF/classes` on Linux to the `bin` folder in the `<JBoss EAP_Home>` folder.
7. Copy the contents of the `classes` folder from `<WebApp_Root>\WEB-INF\classes` on Windows or `<WebApp_Root>/WEB-INF/classes` on Linux to a temporary location (for example, `Temp-Loc`).

Run the following command at `Temp-Loc` to create a `web-inf-classes` JAR file:

```
jar -cvf web-inf-classes.jar *
```

8. Copy the `web-inf-classes.jar` file to `<WebApp_Root>\WEB-INF\lib` on Windows or `<WebApp_Root>/WEB-INF/lib` on Linux.

9. Delete the classes folder from `<WebApp_Root>\WEB-INF` on Windows or `<WebApp_Root>/WEB-INF` on Linux.
10. Disable the tag pooling. To disable the tag pooling, open the `standalone.xml` file in `<JBoss EAP_Home>\standalone\configuration` on Windows or `<JBoss EAP_Home>/standalone/configuration` on Linux in an editor, search for the `<jsp-config/>` tag, update the tag to `<jsp-config tag-pooling=false/>`, and then save the file.
11. Add the configuration entry (in bold) to the `<subsystem>` tag in `standalone.xml` in `<JBoss Home>\standalone\configuration` (Windows) and `<JBoss Home>/standalone/configuration` (Linux) to disable tag pooling:

```
<subsystem xmlns="urn:jboss:domain:web:2.1"
  default-virtual-server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
  <alias name="localhost"/>
  <alias name="example.com"/>
  </virtual-server>
  <configuration>
  <jsp-config tag-pooling="false"/>
  </configuration>
  </subsystem>
```

12. Configure the binding address by replacing `127.0.0.1` with the application server host IP address in the `<wsdl-host>` and `<interfaces>` tags in the `standalone.xml` file.
13. Run the following command at `<WebApp_Root>` to repackage the Records Client WAR file:

```
jar -cvf records.war *
```



Note: When deploying Records Client in JBoss EAP, add `<path name="com/sun/jndi/url/rmi"/>` to `<JBoss EAP deployment folder>\modules\system\layers\base\sun\jdk\main\module.xml` in the `paths` tag.

5.4.2 Classloader settings



Note: If you find any access denied security related errors while using the Records Client, perform the following steps:

1. Create a `permissions.xml` file.
2. In `permissions.xml` file, copy the following content:

```
<?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>
```

3. Place the file in <Tomcat_Home>\webapps\records\META-INF.
4. Restart the application server.

5.4.3 Supporting JDK with JBoss EAP <version> and Elytron security update

JDK is supported with Red Hat JBoss EAP <version> in which the legacy security approach has been disabled and replaced with the Elytron security framework. Therefore, to use JDK, you must update your JBoss environment and migrate the legacy security scripts to Elytron. See the product *Release Notes*, available on My Support (<https://support.opentext.com>), for the supported JDK and JBoss versions.

Updating JBoss environment

1. Download the Red Hat JBoss EAP <version> from the Red Hat Customer Portal.
2. To apply the update, run the following command from JBOSS_HOME:

Windows

```
bin\jboss-cli.bat "patch apply path\to\jboss-eap-<version>-patch.zip"
```

Linux

```
bin/jboss-cli.sh "patch apply path/to/jboss-eap-<version>-patch.zip"
```

Configuring the Elytron security update

1. Back up the configuration directory that contains the legacy security-based configuration.
2. Run the following command:

Windows

```
bin\jboss-cli.bat --file=docs/examples/enable-elytron-se17.cli
```

Linux

```
bin/jboss-cli.sh --file=docs/examples/enable-elytron-se17.cli
```

3. Make the necessary deployment-related changes such as configuring URL or IP, port, and so on in the JBoss `standalone.xml` located in the %JBoss_home%/standalone/configuration folder.
4. Navigate to WEB-INF folder and create the XML file named `jboss-deployment-structure.xml` and add the following tags to the file:

```
<?xml version="1.0"?>
<jboss-deployment-structure>
<deployment>
  <exclude-subsystems>
    <subsystem name="jaxrs"/>
    <subsystem name="webservices"/>
  </exclude-subsystems>
  <exclusions>
    <module name="org.apache.logging.log4j.api"/>
  </exclusions>
</deployment>
</jboss-deployment-structure>
```

Copying the keystore files

1. If available, delete the `dfc.keystore` and `wdk.keystore` files from the *<JBoss Home>\bin* (Windows) or *<JBoss Home>/bin* (Linux) folder.
2. Download and extract the `records.war` file to a separate folder.
3. Copy the `dfc.keystore` and `wdk.keystore` files from *<WebApp Root>\WEB-INF\classes* (Windows) or *<WebApp Root>/WEB-INF/classes* (Linux) folder to the *<JBoss Home>\bin* (Windows) or *<JBoss Home>/bin* (Linux) folder that is generated using the `TrustedAuthenticatorTool` by updating the `dfc.properties` file present in *<WebApp Root>/WEB-INF/classes*.
4. Update the repository details and encrypted password in the *<WebApp Root>\wdk\app.xml* file
5. Configure the binding address by replacing 127.0.0.1 with the ApplicationServer host IP address in the `<wsdl-host>` and `<interfaces>` tags in `standalone.xml`.

Disabling tag pooling and configuring role permissions

1. To disable tag pooling, in `standalone.xml` in *<JBoss Home>\standalone\configuration* (Windows) or *<JBoss Home>/standalone/configuration* (Linux), add the following configuration entry (in bold):

```
<subsystem xmlns="urn:jboss:domain:web:2.1" default-virtual-server="default-host"
native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
  <configuration>
    <jsp-config tag-pooling="false"/>
  </configuration>
</subsystem>
```

2. Navigate to the *<WebApp Root>\WEB-INF* folder and create an XML file named `jboss-deployment-structure.xml`. Add the following tags to the file:

```
<?xml version="1.0"?>
<jboss-deployment-structure>
  <deployment>
    <exclude-subsystems>
      <subsystem name="jaxrs"/>
      <subsystem name="webservices"/>
      <subsystem name="logging"/>
    </exclude-subsystems>
  </deployment>
</jboss-deployment-structure>
```

```

        </exclude-subsystems>
        <exclusions>
            <module name="org.apache.logging.log4j.api"/>
        </exclusions>
    </deployment>
</jboss-deployment-structure>

```

3. Navigate to the *<WebApp Root>/Meta-Inf* folder and create the XML file named *permissions.xml*. Add the following tags to the file:

```

<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>

```

4. Run the following command from the *<WebApp Root>* command prompt to repackage the *records.war* file:

```
jar -cvf records.war *
```

5. Copy the archived *records.war* folder and place it in the *<JBoss_Home>\standalone\deployments* folder and start the JBoss server using the *standalone.bat* file.
6. After the JBOSS server is up and running, approve the Foundation Java API privileges using Documentum Administrator.

5.5 Preparing WildFly

The product *Release Notes*, available on My Support (<https://support.opentext.com>), contains information on the supported versions of WildFly.

5.5.1 Configuring WildFly on Windows

1. Copy the *records.war* file to a temporary location (for example, *<WebApp Root>*).
2. Extract the contents of the *records.war* file using the following command:

```
jar -xvf records.war
```

3. Update *dfc.properties* with the repository details available at *<WebApp Root>\WEB-INF\classes* location.
4. Generate the encrypted passwords for the users 'dmc_wdk_presets_owner' and 'dmc_wdk_preferences_owner' and update the passwords in *app.xml* available at this *<WebApp Root>\wdk* location.

- If Vault is enabled, then the encrypted password is retrieved from Vault.

- If Vault is not enabled, then generate the encrypted password.
5. Perform the following changes in Records Client:
 1. If available, delete the `dfc.keystore` and `wdk.keystore` files in WildFly Home \bin.
 2. Copy the contents of the classes folder from `<WebApp Root>\WEB-INF\classes` to a temporary location, say `<WebApp Root>` and create a jar file `web-inf-classes.jar` using the following command executed at the path `<WebApp Root>`:


```
jar -cvf web-inf-classes.jar *
```
 3. Copy the `web-inf-classes.jar` file to `<WebApp Root>\WEB-INF\classes\lib`.
 4. Delete the classes folder under `<WebApp Root>\WEB-INF\`.
 5. Configure the binding address by replacing 127.0.0.1 with the application server host IP address in `<wsdl-host>` and `<interfaces>` tags in `standalone.xml`
 6. Make the following change in `module.xml` file located at `\<WildFly Home>\modules\system\layers\base\sun\jdk\main` by adding this entry:


```
<path name="sun/security/tools/keytool"/>
```
 7. Repackage the Records war file using the following command executed at `<WebApp Root>` and store the repackaged WAR to another location:


```
jar -cvf records.war *
```
 6. Restart the WildFly server.
 7. Log in to the WildFly management console by accessing the `http://<AppServer-host>:9990/console` link using administrative user credentials.
 8. Add the configuration entry (in bold) to the subsystem tag in the `standalone.xml` file in `<WildFly Home>\standalone\configuration` (Windows) and `<WildFly Home>\standalone\configuration` (Linux) to disable tag pooling:

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"><buffer-cache name="default" />
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https" enable-
http2="true" />
<https-listener name="https" socket-binding="https" security-
realm="ApplicationRealm" enable-http2="true" />
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/><filter-ref name="server-header" />
<filter-ref name="x-powered-by-header" />
<http-invoker security-realm="ApplicationRealm" />
</host>
</server><servlet-container name="default">
<b><jsp-config tag-pooling="false" /></b><websockets />
</servlet-container><handlers>
<file name="welcome-content" path="{jboss.home.dir}/welcome-content" />
</handlers>
<filters>
<response-header name="server-header" header-name="Server" header-value="JBoss-
EAP/7" />
<response-header name="x-powered-by-header" header-name="X-Powered-By" header-
```

```
value="Undertow/1" />
</filters>
</subsystem>
```

9. To configure the Jakarta-supported version of WildFly to work with JDK, navigate to %WILDFLY_HOME%\bin and open the standalone.conf.bat file in a Text Editor.
10. Add the following JAVA_OPTS setting and save the file:

```
set "JAVA_OPTS=%JAVA_OPTS% -Djava.locale.providers=COMPAT,SPI --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"
```

11. To verify the deployment, access the URL. For example, http:<AppServer-host>: 8080/records.

5.5.2 Classloader settings



Note: If you find any access denied security related errors while using the Records Client, perform the following steps:

1. Create a permissions.xml file.
2. In permissions.xml file, copy the following content:

```
<?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>
```

3. Place the file in <Tomcat_install_dir>\webapps\records\META-INF.
4. Restart the application server.

5.6 Preparing Tomcat

The product *Release Notes*, available on My Support (<https://support.opentext.com>), contains information on the supported versions of Tomcat.

5.6.1 Configuring Tomcat for JDK

Add the following parameters in the `catalina.bat` file, located in the `<Tomcat_installldir>/bin` folder, for the JDK `<version>` support:

```
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"
```

5.6.2 Disabling tag reuse

You must disable tag reuse in `/conf/web.xml`. Find the JSP servlet entry and set the `enablePooling` initialization parameter to `false` (as shown in bold):

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>enablePooling</param-name>
    <param-value>false</param-value>
  </init-param>
  <init-param>
    <param-name>fork</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>
```

5.6.3 Disabling HttpOnly property

1. Modify the `<Context>` element in the `context.xml` file located at `<Tomcat_installldir>\conf:`

From

```
<Context>
```

To

```
<Context useHttpOnly="false">
```

2. Restart the application server.

5.6.4 Disabling whitespace parsing

To disable whitespace parsing, add the following line in the `catalina.properties` file located at `<Tomcat_installldir>\conf:`

```
org.apache.jasper.compiler.Parser.STRICT_WHITESPACE=false
```

5.7 Preparing IBM WebSphere Liberty

Records Manager supports IBM WebSphere Liberty. *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* contains detailed information.

5.7.1 Configuring for the Jakarta-supported version of IBM Websphere Liberty

To use the Jakarta-supported version of the IBM Liberty server, you must add the following entry in the `server.xml` file in the application server:

```
<featureManager>
<feature>jakartaee-10.0</feature>
</featureManager>
```

You must also update the host IP address and port in the `<httpEndpoint>` tag and then deploy.

For example:

```
<httpEndpoint id="defaultHttpEndpoint"
httpPort="9081"
httpsPort="9444"
host="10.194.53.195" />
```

To configure IBM Websphere Liberty for JDK <version>:

1. Navigate to `<IBM_Liberty_installldir>\wlp-jakartaee<version>\wlp\bin\` and open the `server.bat` file.
2. Add the following `JAVA_OPTS` setting and save the file:

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

5.8 Preparing to use an external web server

Refer to the product *Release Notes* on My Support (<https://support.opentext.com>), to determine whether external web servers are supported for your product.

External web servers are sometimes used as a front end to the application server. For example, an external web server may be used for balancing the loads on a collection of application servers or used as a forward or reverse proxy server.

UCF content transfer uses chunked transfer encoding, a standard of the HTTP 1.1 specification. Many proxy web servers such as Sun Server implement chunked transfer encoding in a way that does not work properly with UCF. If the external server does not support HTTP 1.1 chunked encoding, you must configure UCF in the WDK-based application to use an alternative chunked encoding. The *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* contains information on this configuration.

When deploying in a manager server or network deployment environment, the external web server must provide session.

Chapter 6

Installing the Records Client products

The Records Client products are named as follows:

OpenText™ Documentum™ Content Management Retention Policy Services

Its administration node in the Records Client navigation pane is also named OpenText Documentum Content Management (CM) Retention Policy Services.

OpenText Documentum Content Management (CM) Records Manager

Its administration node in the Records Client navigation pane is also named Records Manager.



Note: Make sure that all Records Client products installed on the same repository are of the same release version. The Records WAR file must be the same version as the version of the Retention Policy Services and Records Manager DAR files installed on the repository.

Be sure that your system meets the requirements listed in the product *Release Notes* that is available on My Support (<https://support.opentext.com>). You must use Records Client 25.4 DAR files such as `rps.dar` and `rm.dar` when installing Records Client 25.4.

When you have OpenText™ Documentum™ Content Management Records Queue Manager installed in your environment, clear the Foundation Java API Business Object Framework (BOF) cache from the installed hosts after applying the Records Client 25.4. For products like Records Adapter for Microsoft SharePoint and Records Activator for Microsoft Outlook, you must use the 16.7 version with Records Client 25.4.

6.1 Overview

This chapter contains product specific information for installing the unified Records Client, and the products it hosts, for the current release. Follow procedures in the applicable chapter for preparing the Documentum CM Server, application server, and client host before proceeding to the primary steps. Refer to the product *Release Notes* on My Support (<https://support.opentext.com>), to prepare them according to the hardware and software requirements.

The Records Client is built on top of Webtop to include core OpenText Documentum CM functionality along with all Records Client product functionality. It is a unified solution that makes it possible to deploy all the Records Client products from a single WAR file. This leaves only the DAR files of the required product to be installed. Retention Policy Services is the base Records Client product that can be used as a standalone solution or combined with the other two as follows:

- Retention Policy Services
- Retention Policy Services + Records Manager

Only basic Webtop functionality is accessible if the Records Client is not installed.

The GUI-based DAR Deployer plug-in for Documentum Composer is available on the download site.

See the product *Release Notes* available on My Support (<https://support.opentext.com>), before proceeding. Use the release notes to ensure your system meets or exceeds environment and system requirements for new deployments or upgrades. If your system components are not supported versions, follow instructions in preceding chapters to prepare the Documentum CM Server, application server, and the client host.

To complete and verify the deployment, after completing all of the major steps, refer to “Completing the deployment” on page 107.

If you encounter any problems, refer to “Troubleshooting the new installation” on page 49.

6.2 Primary steps

6.2.1 Download the Records Client files

This section provides details of all the files that you need to download for your combination of records deployment.

The *records.war* file is the unified **Records Client** and is mandatory for any records deployment, regardless of the records products purchased. You can download this file from My Support.

Download the *records.war* file to the application server and also download the respective DARs, to the Documentum Server, according to “DAR file download matrix” on page 38, for your combination of records deployment. When downloading the various files, also make sure to download any language packs if support for a particular language other than English is required.

For DAR file installation instructions, follow the primary steps for creating both the global and Records Client (main) repositories. However, to install a particular language pack DAR and to apply it, refer to the following sections in “Deploying language packs” on page 17:

- Installing the DAR with localized data dictionary properties files
- Applying language packs

The first procedure provides instructions to install the DAR against the repository, the second provides the WAR file deployment instructions against the application server. It does not matter which is done first if separate servers are used. It is

recommended to install the DAR Language Pack with the corresponding product DAR at the same time. The DAR deployer includes an option you can select to install the language pack DAR when you install a product DAR. If you do not install it at the same time, and decide to install it at a later time, you will have to re-install the product DAR to select this option.

Before installing the product DAR with the localized data dictionary for the desired language, enable the required locale in the repository. When a repository is created, a set of data dictionary information is loaded, based on the Documentum CM Server host locale. If the host locale is Russian, Arabic, Japanese, Korean, or Simplified Chinese, the data dictionary information for that locale is loaded during repository creation. Otherwise, the host locale is always English. To add a locale, use the population script provided by OpenText Documentum CM. You can populate and publish the data dictionary file by following the procedures located in the **Populating and Publishing the Data Dictionary** appendix in the *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)*.



Note: To avoid DAR deployer issues, it is recommended that you store DAR files associated to a particular product in the same folder. Storing all of the DAR files of each of the products in the same folder is okay. However, if you are storing DARs in different folders, keep associated DARs together in the same folder. This will prevent installer issues, in particular when the installer is used at a later time to install an optional DAR that has a dependency on another DAR. The Records Manager DAR for example, cannot be installed successfully if the installer cannot find it in the same folder from which the Retention Policy Services DAR was installed.

Refer to the matrix in [“DAR file download matrix” on page 38](#) to determine which DAR files to download for the combination of records functionality desired.

For example, if you are entitled for only Retention Policy Services, you need to download only the `rps.dar` file.

From 25.2 onwards, the following DAR files have been combined into the single Records Manager DAR, `rm.dar`:

- `rm.dar`: Installs Records Manager. This is the main DAR for Records Manager.
- `prm.dar`: Installs Physical Records Manager. If for any reason the Retention Policy Services DAR file has to be reinstalled on installations where Physical Records Manager is installed, make sure that the Records Manager DAR file is also reinstalled immediately after the Retention Policy Services DAR file is installed.
- `rmce.dar`: Installs Documentum Records Manager Commonwealth Edition (RMCE), which is required for creating Files, File Parts, and records in accordance to RMCE.
- `RM-Forms-Adaptor.dar`: Deploys classes required for creating formal records.
- `RM-Default.dar`: Provides sample form templates for formal records, formal folders, and formal cabinets.

- `RM-DoD5015v3-Standard-Record.dar`: Installs the Department of Defense (DoD) specific to version 3 records functionality. This functionality is required only if DoD version 3 standard records or DoD version 3 email records are to be declared.




Note: The export directory for email records must be specified on the Java Method Server. The Java Method Server must be modified with the export directory to support email export functionality when the disposition job attempts to export email records. Also, note that the Disposition job will not process disposition strategies that include a transfer strategy unless the Java Method Server is running on a Windows machine. Refer to exporting section to specify an export directory on the Java Method Server (JVM).

- `RM-DoD5015v3-Classified-Record.dar`: Installs the DoD specific to version 3 records functionality. This functionality is required only if DoD version 3 classified records are to be declared.
- `RM-DoD5015-2.dar`: Installs the DoD specific to version 2 records functionality that provides support for the earlier DoD version 2 records and assures that they will function properly in the new system. It does not migrate these records into the new version 3 types however.
- `RM-DoD5015-4.dar`: Installs the DoD specific to version 2 records functionality that provides support for the earlier DoD version 2 records and assures that they will function properly in the new system. It does not migrate these records into the new version 3 types however.

Table 6-1: DAR file download matrix

DAR install order	DAR name	Install DAR to the repository	Download Area		Notes
			Retention Policy Services	Records Manager	
1	<code>rps.dar</code>	Main	x	x	This is the main DAR for Retention Policy Services. Make sure to manually resolve Pending Policies, if there are any, before installing <code>rps.dar</code> .
2	<code>Forms.dar</code>	Global and Main		x	This DAR is required for Records Manager. This DAR installs the forms framework for the repository.
3	<code>rm.dar</code>	Main	x	x	This is the main DAR for Records Manager.




DAR install order	DAR name	Install DAR to the repository	Download Area		Notes
			Retention Policy Services	Records Manager	

 **Note:** A Documentum Composer reference project file for example: `rmce_ReferenceProject.zip`, `rm_ReferenceProject.zip`, `RM-Default_ReferenceProject.zip`, and so forth, is included for each of the respective DAR files. You need the respective reference project file only if you plan to sub type object types from the artifacts of a particular DAR. Documentum Composer allows you to create references between projects to share resources such as OpenText Documentum CM artifacts, libraries, or Java ARchive (JAR) files. If any are needed, extract it and import it into Documentum Composer to run it.

The matrix of miscellaneous files in [“Miscellaneous files download matrix” on page 39](#) are optional downloads.

Table 6-2: Miscellaneous files download matrix

File Name	Download Area		Notes
	Retention Policy Services	Records Manager	
<code>recordsServiceJavaDocs_25.4.zip</code>	x	x	Javadocs for the Java public API for records.
<code>rps-Webservices.zip</code>	x		This is the package for installing Web Services for Retention Policy Services. The Records Manager web services is also included.
<code>rm-webservices-25.4.zip</code>	x	x	This is the package for installing web services for Records Manager. The web services include the web services for Retention Policy Services and Records Manager. Therefore, there is no need to deploy the Retention Policy Services web services if this is deployed.

File Name	Download Area		Notes
	Retention Policy Services	Records Manager	
dod_certification.zip		x	<p>This is an optional package that contains 5 utilities (follow instructions included in the README file in each directory):</p> <ul style="list-style-type: none"> ContentServerFileChecker This tool checks the checksum of every sysobject in the repository to ensure its integrity. <p> Tip: Do not run this utility against a repository that has a large number (millions) of sysobjects. Otherwise, this will run for a very long time consuming Documentum CM Server resources.</p> Delete Group This is a docbasic ebs script that allows an administrator to delete a group. It will remove all references from existing ACLs and groups. If a customer is referencing the group from their own custom attribute, they will need to update there objects. <p> Caution This operation cannot be undone.</p> Group Renaming Utility This is a docbasic ebs script that allows an administrator to rename a group. It will replace all references of the group to the names for ACLs and groups. <p> Note: The new group name should not already exist.</p> MakeClassificationGuideFirstInSearchColumns This is an optional customization that forces the classification guide to be first in search columns and prevents end users from changing the first column. NamingPolicy Instructions on how to define the naming policy that is used for the naming and

File Name	Download Area		Notes
	Retention Policy Services	Records Manager	
			construction policy for DoD testing. Useful for non-DoD as an example.

6.2.2 Creating the global repository

When an OpenText Documentum CM installation includes multiple repositories, certain installation-wide elements are shared among all repositories. To manage these installation-wide elements, each installation includes a central repository called the global registry, also known as the global repository.

A global repository is a mandatory requirement for records functionality, refer to [“Required and optional supporting software” on page 11](#) for more details. Although the global repository and the main repository can be the same repository, OpenText Documentum CM strongly recommends creating them separately.

Create a global repository on the Documentum CM Server for shared access to privileged code (Privileged DFC), if one has not already been created. Then, only if you are entitled for Records Manager however, install `rps.dar`, `Forms.dar`, and `rm.dar` using Documentum Composer, according to the DAR Install Order in [“DAR file download matrix” on page 38](#).



Note: `Forms.dar` must be installed on both the main repository for which Records Manager is available and on the global repository. If more than one main repository is involved, install it on all of them.

6.2.3 Creating the Records Client repository

Create the main repository, that is the application repository for the Records Client. For records to work properly, that is, to avoid unexpected behaviors, ensure that the latest versions of Records Client, Documentum CM Server, and DARs are installed.



Note: If a separate global repository is used, the DAR versions for the Records Adapter must match the version of the repository. If the customer uses more than one repository, all versions must have the same DAR versions installed (including patches).

6.2.4 Installing the DAR files with the DAR Deployer plug-in

You can use the DAR Deployer plug-in to install a DAR file to the repository if you do not want to use the interface within Documentum Composer. The DAR Deployer plug-in requires Documentum Composer to be installed, but does not launch the full Documentum Composer IDE. The DAR Deployer Plug-in is useful in cases where you want to decouple the development of DAR files from the installation of DAR files. It is also useful in situations where the installation of DAR files is required as part of a deployment process. When you open the DAR Deployer plug-in, it creates three folders in your Documentum Composer installation directory:

- darinstallerconfig - contains configuration files for the DAR Deployer program
- darinstallerlogs - the default location of the log files
- darinstallerworkspaces - workspaces that are created and used by the DAR Deployer program. The DAR Deployer program does not delete these workspaces automatically, so you occasionally need to clean up this directory. The workspace directories are named in the following form: darinstaller_workspace_yyyy-mm-dd-hh-mm-ss.

The DAR Deployer requires you to fill in certain values that are marked with an asterisk (*). All other fields are optional. For a description of the fields for the DAR Deployer Utility, refer to [“Description of DAR Deployer fields” on page 44](#).



Note: To add support for other languages, that is for non-English locales, you need to specify the desired DAR language pack and reference the product *Release Notes*, to install it.

If you receive an error that the DAR file exceeds the maximum size, you need to change the allowed size. To do so:

1. Locate C:\Program Files\Apache Software Foundation\<Tomcat version>\webapps\manager\WEB-INF
2. Open the web.xml file and locate the Manager servlet multipart-config parameter.
3. Change the max-file-size and max-request-size to 390000000 so that it looks as such:

```
<multipart-config>
  <!-- 50MB max -->
  <max-file-size>390000000</max-file-size>
  <max-request-size>390000000</max-request-size>
  <file-size-threshold>0</file-size-threshold>
</multipart-config>
```

4. Save and restart Tomcat.

To install a DAR file:

1. Download and unzip the DAR Deployer zip file to the root of your Documentum Composer or Headless Composer installation directory.

You can find the DAR Deployer plug-in from the Documentum Composer product binary.

It should be noted that Documentum CM Server comes equipped with Documentum Composer already installed for you. Refer to the applicable bullet item that represents your setup to install Documentum Composer or Headless Composer if it is not already installed on your version of Documentum CM Server or other host:

- The built-in Documentum Composer, installed on the Documentum CM Server is found in `C:\Documentum\product\<version>\install\composer\ComposerHeadless`.
- The standalone Documentum Composer, installed on a host other than the Documentum CM Server is found in `C:\Composer_tools\DCTM_Headless_Composer`.



Note: Although either Documentum Composer or Headless Composer needs to be installed for the DAR Deployer plug-in, it also needs to be configured for the applicable connection broker host and port.

Follow these substeps, also documented in the *OpenText Documentum Content Management - Composer User Guide (EDCPC-UGD)*, to configure the connection broker, regardless of whether Documentum Composer or Headless Composer is built-in or standalone.

- On your local machine, change to the `..\Composer\plugins` directory.
- Double-click the `com.emc.ide.external.dfc_1.0.0` folder.
- Double-click the `documentum.config` folder.
- Open the `dfc.properties` file with a text editor, such as Notepad. Add the Foundation Java API and connection broker information, similar to the following:

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

- Start the DAR Deployer Program, **dardeployer.exe**.
- In the **DAR Details** section, specify values for the fields. Make sure that when you browse to select a DAR to install, that you follow the DAR Install Order in **“DAR file download matrix” on page 38** based on the Records Client products installed.
- In the **Docbroker Details** section, specify values for **Docbroker Host** and **Docbroker Port** and click **Connect**.
- In the **Repository Details** section, specify values for the fields and click **Install** to install the selected DAR file to the repository.

Refer to **“Description of DAR Deployer fields” on page 44** to help guide you through the DAR file install process.

You can view the log for the DAR installation by selecting the log file from the **Log File** drop down menu and clicking **Open**.

Table 6-3: Description of DAR Deployer fields

Parameter	Required	Description
DAR	Yes	The absolute file path to the .DAR file that you want to install. The file path cannot contain any I18N characters or the installation will fail.
Input File	No	The input file can be used to change the default filestore used to deploy the application to. Additional details are in the Documentum Composer documentation.
Local Folder	No	The absolute file path to localized .properties files. If you want to make your application available in other languages, you need to localize the project data such as labels, tabs, and descriptions.
Log File	No	The file to save the log to. If this is not specified, the file defaults to <DAR>.log.
Docbroker Host	Yes	The address of the connection broker.
Docbroker Port	Yes	The port of the connection broker repository.
Repository	Yes	The name of the repository that you want to install the DAR file to. Click the Connect button after entering the connection broker host and port to retrieve the available repositories.
User Name	Yes	The login name for the repository.
Password	Yes	The password for logging into the repository.
Domain	No	The domain where the repository resides.

If you happen to install a DAR file in the wrong order, simply close and reselect the correct DAR file.



Note: To avoid having to reinstall a DAR, due to an upgrade for example, store all product related DARs in the same directory. Although all of the DARs can be stored in one directory, make sure you store all of the DARs associated to a particular product in the same directory.

Refer to the latest *OpenText Documentum Content Management - Composer User Guide (EDCPC-UGD)* to learn more about Documentum Composer.

Special instructions:

1. Clear the BOF cache found in the `dfc.data.dir`. The path of `dfc.data.dir` can be found in `dfc.properties` of the Documentum CM Server on Windows installations. Typically, it is found at `C:\Documentum\config` on the Documentum CM Server. If it is not possible to empty the cache due to some process holding it:
 - Edit `dfc.data.dir` of `dfc.properties` to point to a different location, to create a new cache.
2. Make sure that `dfc.properties` of the DAR Deployer is identical to `dfc.properties` of the Documentum CM Server. The `dfc.properties` of the DAR deployer is typically found at `C:\Documentum\product\<version>\install\composer\ComposerHeadless\plugins\com.emc.ide.external.dfc_1.0.0\documentum.config`.

6.2.5 Installing the DAR files using Headless Composer on Linux

For installing any records DAR file with Headless Composer on Linux operating systems, you should first install all its reference DAR files in the same order as referenced and then install the specific records DAR file. For this, create your own Ant script that contains targets to install the reference DAR files first and then install the specific records DAR file. These targets must be in the same Ant script and must be run in the same call to Ant.

For example, `rm.dar` references `rps.dar`. So, for installing `rm.dar` with Headless Composer on Linux operating systems, you should first install `rps.dar` and then install `rm.dar` ensuring that these Ant targets are all invoked from the same Ant script and also through the same call to Ant.

The *OpenText Documentum Content Management - Composer User Guide (EDCPC-UGD)* contains more information.

6.2.6 Configuring Records Client to connect to the Vault server

1. Perform all the steps as described in “[Configuring Documentum Secret Integration Service](#)” on page 14.
2. Open the `dfc.properties` file and add the following lines:

```
dfc.dsis.enabled=true
dfc.dsis.daemon.url=http://localhost:8200/dsis
dfc.dsis.daemon.token=<token generated for the DSIS daemon agent>
```

For more information about setting the preceding entries in the `dfc.properties` file, see *OpenText Documentum Content Management - Foundation Java API Development Guide (EDCPKCL-DGD)*.

3. Redeploy the WAR file.
4. Restart the application server.

6.2.7 Installing records.war on the application server



Warning

Make sure to manually resolve Pending Policies, if there are any, before upgrading the records.war.



Note: If an error is displayed indicating that the war file exceeds the maximum size, change the allowed size. To change the allowed size, navigate to C:\Apache Software Foundation\<Tomcat version>\webapps\manager\WEB-INF. Open the web.xml file and locate the Manager servlet multipart-config parameter. Change the max-file-size and max-request-size to 390000000 as follows:

```
<multipart-config>
  <!-- 50MB max -->
  <max-file-size>390000000</max-file-size>
  <max-request-size>390000000</max-request-size>
  <file-size-threshold>0</file-size-threshold>
</multipart-config>
```

Save and restart Tomcat.

1. Ensure the server is the supported version according to the product *Release Notes* that is available on My Support (<https://support.opentext.com>).
2. Verify that you can access the Test Page/Admin Page for the application server.
3. Install the records.war file on the application server. The records.war file can be deployed with or without a GUI.



Note: Only the records.war file is necessary as it will display all Retention Policy Services and Records Manager functions if these products are properly installed. For application servers that use WARS, if you want to use a different context-root, rename the records WAR file from records.war to whatever context-root you prefer to use. For example, the default URL is `http://<hostname>:<port>/records`. If you want to use `http://<hostname>:<port>/rma` (as in earlier releases), rename the war file to rma.war.

Refer to “Preparing the WAR file for deployment” on page 97 for additional instructions.

To install the records.war without the GUI, refer to the deployment notes for your application server.

4. After the deployment, add the following information as formatted to the dfc.properties file under \${<deployed application path>}/WEB-INF/classes.

You can obtain the values needed for the Records Client dfc.properties file from the Documentum CM Server dfc.properties.

```
dfc.docbroker.host[0]=
dfc.globalregistry.repository=
dfc.globalregistry.password=
dfc.globalregistry.username=
```

- If Vault is enabled, then the password is retrieved from Vault.
 - If Vault is not enabled, then you must provide the password.
5. Disable tag pooling and also configure memory allocation. For more information, see [“Preparing the application server host” on page 23](#).
 6. Restart the application server.
 7. Log in to Records Client from a client machine browser (http://Host_Name:Port_Number/Context_Root). Logout immediately, after authentication.



Note: The reason for this step is for the Records Client to populate its identify to the global repository. If this is not done, you cannot register the client as the system will not know its identity (stored in the `dfc.keystore`).

6.2.8 Importing WebFormService.jar into Documentum CM Server

For formal records to work, `WebFormService.jar` must be imported into the Documentum CM Server repository:

1. Navigate to the `<Records_install_dir>\WEB-INF\HelperJars` location on the application server.
2. Copy the `WebFormService.jar` to a temporary folder in Documentum CM Server. For example, `C:\temp`.
3. Import the JAR file to the Documentum CM Server repository by running the following IAPI query:

```
retrieve,c,dmc_jar where object_name = 'WebFormService.jar'
checkout,c,l
setfile,c,l,C:\temp\WebFormService.jar,jar
checkin,c,l
```

For more information on importing JARs, see Documentum CM Server documentation.

4. Proceed to [“Setting up Documentum Administrator” on page 48](#).



Note: To register this instance for Privileged DFC, refer to [“Registering all Foundation Java API clients” on page 48](#) for instructions.

6.2.9 Setting up Documentum Administrator

Install Documentum Administrator, if it is not already installed, to register the clients for privilege. To set up Documentum Administrator, refer to the version of the *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS-AGD)* that is available for the current release.

6.2.10 Registering all Foundation Java API clients

Any Foundation Java API client that connects to a Records Client repository for the first time must be registered for Foundation Java API according to these instructions; this includes Records Manager and Retention Policy Services Web Services.

To register a client:

1. Log in to Documentum Administrator as the install owner.
2. Click on the Privileged Clients node to register Foundation Java API.
3. If you see your application server and Documentum CM Server Foundation Java API instances displayed then select them and right-click and select Approve Privilege from the listbox.

If you do not see your application server and Documentum CM Server Foundation Java API instances displayed, then click **Manage Clients** under **File**. You will see your application server and Documentum CM Server instances on the left screen. Select them and move them to the right screen. Click **OK**. Select the two Foundation Java API instances and right-click **Approve Privilege**.

If you do not see your application server instances, remove the `dfc.keystore` file on the application server and restart the server. Log in to Records Client again and the Foundation Java API instance should appear.
4. Log in to Foundation Java API and try to create a security policy from the Records Manager node and a contact from the Retention Policy Services node. If you get an error during creation, you may not have configured Privileged DFC correctly or the Foundation Java API instance registered may not be the correct one.

6.3 Configuring users, roles, and audits

The *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* provides information on how to configure users, roles, and audits and how to register various audit events along with a complete list of all the audit events.

6.4 Troubleshooting the new installation

Refer to these tips/instructions if you encounter the following problems:

- OpenText Documentum CM services may not restart properly when you restart the Documentum CM Server. Perform the following steps if one or more of the services are not running:
 1. Stop the OpenText Documentum CM services that are currently running.
 2. Start the services in the following order:
 - a. Documentum Docbroker Service
 - b. Documentum Docbase Service
 - c. Documentum Java Method Server
- Click the **Stop** button then the **Start** button to restart the service.
- Check the DAR install log for DAR files installed in c:\temp\documentum. The log files start with the dcm_tools_ prefix.
- Check that the application build install folder contains XX_install.xml.
- On the Documentum CM Server, perform the following steps if you reinstall the DAR files:
 1. Stop OpenText Documentum CM services that are currently running.
 2. Clear the Foundation Java API BOF cache in c:\documentum\config folder or in dfc.data.dir path folder.
 3. Start the services in the order listed in 2 on page 49.
 4. Reinstall the DAR file.
 5. Check the error log against the DAR files installed in c:\temp\documentum.
- If the install log contains an authentication error make sure the connection broker can be connected.
- If you get the following error post installation:

```
DM_SESSION_E_SETUP_ROLES_FOR_RPC]error: "Error initializing dynamic roles for RPC
RelationSave (379c20cc80086d5c)"; ERRORCODE: 100; NEXT: DfException:: THREAD:
[ACTIVE] ExecuteThread: '26' for queue: '<app server>.kernel.Default (self-
tuning)'; MSG: [DM_GROUP_E_PROTECTED_ROLE_UNKNOWN_CLIENT]error: The requested role
(dmc_rec_escalated_full_control) is marked.
```

Then, on the Retention Policy Services diagnostics page, <http://Application url:port/records/component/diagnostics>, register and approve the listed Foundation Java API client.

6.5 Upgrading from earlier releases

6.5.1 Upgrade paths and overview

If you plan to upgrade, refer to the product *Release Notes* available on My Support (<https://support.opentext.com>), before proceeding. Use the release notes to determine if there are any bugs that can impact the installation and in particular, to ensure that your system meets or exceeds environment and system requirements. Also, refer to “[Upgrading a WDK-based application](#)” on [page 95](#) to perform preliminary backup tasks before proceeding.

Refer to “[Upgrading and migrating previous versions of the Records Client to the latest versions](#)” on [page 50](#) to determine the upgrade path from an earlier version of one or more of the Records Client products to a newer version. Only one upgrade is required; all the Records Client products can be upgraded directly to 25.4. For example, it is not necessary to upgrade Retention Policy Services 20.4 incrementally to 21.2, 21.4, 22.1, and then to 25.4. To determine which version is currently installed, log in to the client and click **File > About Records Client**.

6.5.2 Upgrading and migrating previous versions of the Records Client to the latest versions

There are three key procedures to follow when you upgrade an existing system:

1. Upgrade the Documentum CM Server to the latest version. However, refer to the product *Release Notes* available on My Support (<https://support.opentext.com>) for the supported versions and decide if upgrade is required.
2. Install the new Records Client DARs for 25.4.
3. Set up the application server.

Make sure to stop using the earlier client after completing an upgrade. Use the new version of the Records Client after the new DARs have been installed.

To upgrade the Documentum CM Server to the latest version:

1. Back up all repositories.
2. Upgrade Documentum CM Server according to the *OpenText Documentum Content Management - On-Premises Upgrade and Migration Guide (EDCCS-UMD)*. These are the key steps:
 - a. Shut down the Java Method Server.
 - b. Shut down the repositories and connection brokers.
 - c. Upgrade Documentum CM Server.
 - d. Upgrade the connection broker and all repositories.
3. Create a global registry repository if one was not already created. Refer to “[Creating the global repository](#)” on [page 41](#) if necessary.

To install the Records Client DARs:

- Refer to “[Installing the DAR files with the DAR Deployer plug-in](#)” on page 42 to install the DARs required for the Records Client products that are installed.

Step 3 of those instructions references the DAR Install Order in “[DAR file download matrix](#)” on page 38 for the products that are installed.

To set up the application server:

1. Uninstall, or remove, the previous version of Retention Policy Services or Physical Records Manager or Records Manager.
2. Install the Records Client according to “[Installing records.war on the application server](#)” on page 46.

6.5.3 Migrating custom record relationship instances

Follow these instructions to migrate custom record relationships defined on prior versions of Records Manager to the current version.

To successfully migrate prior versions of custom record relationship instances:

1. Create your own relation type, using Documentum Composer, one for each custom record relation type to be migrated.

To do this, refer to *OpenText Documentum Content Management - Composer User Guide (EDCPC-UGD)*. The instructions for *Creating a relation type* can be found in the chapter *Managing Relation Types*.

The relation types define the parent and child object types that can be related. For example, if the relation type defines the allowable parent object type as `dm_document` and the allowable child object type is also defined as `dm_document`, only those objects that are `dm_document` types can be selected when a relationship is created. Record relation definitions can only be created, according to step 3, after the relation types in this step and the object types according to step 2 are created. Various record relationships can then be created from the record relation definitions created.

Properties: Record Relation Definition

Info History

Suspend Relationship

Type: dmc_rm_record_relation_def

Format:

* Name: Suspend Relationship

* Relation Object Type: Suspend Record Relation (dmc_rm_suspend_rel) ▼

Relation Type

* Relation Name: dmc_rm_rec_suspend_rel_type ▼

Description: Suspending Record Relation

Security Type: NONE

Allowable Parent Type: dm_document

Allowable Child Type: dm_document

Parent Relation Label: Is Suspending

Child Relation Label: Is Suspended By

Relationship Permissions

* Members that can Create: dmc_rm_privilegeduser [Edit](#)

* Members that can Remove: dmc_rm_recordsmanager [Edit](#)

? OK Cancel

Figure 6-1: Relation type example of allowable parent and child type for creating relationships

2. Optionally, create the relation object type (sub-type of dm_relation) using Documentum Composer, for each relation type created in step 1.
 To do this, refer to *OpenText Documentum Content Management - Composer User Guide (EDCPC-UGD)*. The instructions for *Creating a standard object type* can be found in the chapter *Managing Types*. If you decide not to create a new object type, use the existing Record Relation (dmc_rm_record_rel) for the Relation Object Type.
3. Create the records relation definitions, using the Records Client, for the desired record relationships. For the **Relation Name** field, select the name of the relation type. For the **Relation Object Type**, use the value from step 2 (so if you do not want to make an new object type, select **Record Relation (dmc_rm_record_rel)**). Fill in the rest of the mandatory fields.

You are now set up to create record relationships from the record relation definitions created. See *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* to create record relationships.

6.6 Configuring security

Records Client supports folder security whether it is turned on or off. If Records Manager is installed and you are using security policies, you can turn it off as Records Manager policies can fine tune security if desired. Folder security, which is on by default, can be set from Documentum Administrator and is further described in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

6.7 Working with the new email processing functionality

See *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* for more details about the email processing functionality.

6.7.1 Working with new email messages imported from Records 20.4 onwards

New email messages are imported as objects of `dm_document` type or its subtypes and stored in the repository in their native Microsoft Outlook (.msg) format itself. Import of new email messages in EMC MF format as objects of `dm_message_archive` type or its subtypes is no more supported.

To work with the new email message functionality:

- You must install `MailApp.dar` on the Documentum CM Server. The product *Release Notes* available on My Support (<https://support.opentext.com>) contains detailed information on the system requirements.
- The `mailapp.properties` packaged with Records WAR must be configured appropriately. “*Configuring the MailApp.properties file*” on page 53 provides detailed information about `mailapp.properties`.

6.7.2 Configuring the MailApp.properties file

By default, `MailApp.properties` is located in `<WebApp Root>\WEB-INF\classes`. You can modify the values of different properties in `mailapp.properties` to change the default behavior. “*Properties of MailApp.Properties file*” on page 54 provides more details about the different properties available in the `mailapp.properties` file.

Table 6-4: Properties of MailApp.Properties file

Property	Default value	Description
shouldParseMsgFile	Property set to true	<p>When this property is set to true, the email message is imported into the repository in .msg format as an object of dm_document type or its subtypes and then email specific functionality is enabled.</p> <p>If this property is set to false, then the email messages will be imported as a normal document.</p>
shouldSeparateAttachments	Property set to false	<p>This property is applicable only when shouldParseMsgFile is true.</p> <p>When this property is set to false, the attachments are not extracted from the email message and they are not stored as separate objects in the repository.</p> <p>If this property is set to true, then all the attachments stored inside the email message are extracted and stored in the repository as separate objects. The attachments are stored inside an attachments folder. This folder is hidden by default.</p>

Property	Default value	Description
objectTypeForAttachments	The value of this property is set to <code>dm_document</code>	<p>This property is applicable only when shouldParseMsgFile and shouldSeparateAttachments properties are set to true.</p> <p>Whenever extracting of attachment is enabled by setting shouldSeparateAttachments property to true, all the non-email type of attachments present inside the email message are extracted and stored in the repository as separate objects of the type.</p> <p>The object type of the nested email attachments is set to the same type that was selected by the user for the main email message during the import operation.</p>



Notes

- Regardless of whether extraction of attachment is enabled or not, the email messages imported in .msg format will always contain all its attachments inside it.
- Whenever attachment extraction is enabled in `mailapp.properties`, you can view the hidden attachments folder by selecting **Tools > Preferences > Show Hidden Files**.

6.7.3 Using configuration options in `app.xml` for email import

Following configuration options (optional) are available in `app.xml` for importing emails in `<mailMessage-support>`.

- `<skip-duplicate-messages>`: Specifies whether to log errors for duplicate messages and continue importing (true) or throw an error and stop importing (false). The default value is `<true>`.
- `<override-object-name>`: Specifies whether you can change the object name of email objects through import. The default value is `<false>`.

6.8 Configuring Records Activator for declaring email records from Microsoft Outlook

To file or declare email as formal records, the Records Activator for Microsoft Outlook must be installed and enabled based on two sets of instructions. The first set of instructions provides steps to install on the application server and the second set provides steps to install on the client machine.

Refer to the product *Release Notes* available on My Support (<https://support.opentext.com>) to determine the supported versions of the following items, before proceeding:

- Supported versions of the Windows operating system
- Supported versions of Outlook
- Supported versions of JDK

Steps to perform on the application server:

1. Set the `shouldParseMsgFile` property in `<WebApp Root>/WEB-INF/classes/mailapp.properties` to **true**.
2. To be able to include attachments while declaring email, set `shouldSeparateAttachments` in `mailapp.properties` to **true**.
3. Set the appropriate value for the `Source Email Type For Filing Email Records` attribute in `Records Manager Docbase Configuration` object to configure the object type of the source email message imported while declaring the email record.



Note: The source email message, which is declared as an email record, is imported into repository in its native outlook message format (.msg format) as an object of `dm_email_message` type or any of its subtypes. For more details, see *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

4. Restart the application server.

Steps to perform on the client machine:

1. Make sure that Outlook is installed and configured to send and receive e-mail.
2. Uninstall any previous versions of Documentum Records Manager Outlook Activator.
3. Make sure that the following Java instances are not running:
 - `java.exe`
 - `javaw.exe`

4. Install the supported version of JRE, if it is not already installed. The product *Release Notes* available on My Support (<https://support.opentext.com>) contains the details.
5. *For Outlook 2016 or later:* Run SetupOutlookRA.exe. If the Regional Language setting in Office 2016 or later is other than English, delete the Registry at location HKEY_CURRENT_USER > Software > Documentum > Outlook if exists and then restart the Outlook. English is the default language.



Note: Ensure that the correct installer is downloaded and used for Outlook.

6. Only if you are upgrading, delete the UCF cache on the client machine, to ensure a new version of UCF is downloaded. That is to delete everything in the folder, %UserProfile%\Documentum\ucf.

Delete all files in the Java cache on the client machine. On Edge browsers, delete all files in the Java cache.

Delete the Webtop Native Client files. You can get the location of the files from the value of the registry entry *Path* at HKEY_CURRENT_USER > SOFTWARE > OpenText > Webtop > NativeSetup.
7. Using the Windows Components Wizard, uninstall *Internet Explorer Enhanced Security Configuration* by clearing the check box. This step is important to prevent blocking the download of the Webtop Native Setup client.
8. Using the browser, log in to the Records Client to force it to install Webtop NativeSetup.exe. In the warning screen, click **Run**, when prompted.
9. Click **Yes** on the Trusted site to confirm, so that the Native Setup client is downloaded.
10. Import a new email message from Records Client to ensure that the new email messages are imported as per the configurations defined in mailapp.properties.
11. Proceed to declare an email record from Records Activator for Microsoft Outlook according to instructions in the *OpenText Documentum Records Activator for Microsoft Outlook - User Guide (EDCRMCOOUT-UGD)*.

6.9 Configuring Records Manager for barcode label printing and batch processing

Refer to this section after installing the Records Manager DARs and only if you want to include the optional label printing capability or batch processing operations or both. Although these are optional, the setup of Records Manager is required for either one.

Records Manager supports barcode label printing and batch processing operations. It is required for printing barcode labels from NiceLabel templates (using NiceLabel 17 or later versions) for physical objects in inventory and to upload barcode readings, either manually through the web browser or automatically from an FTP folder (which is also called the Physical Records Manager Auto-Queue). All physical objects using these two features can be barcoded and then when scanned, according to a batch processing operation, can be uploaded to update the inventory database.



Note: From the NiceLabel software suite, you need only Nice Label.

The initial installation requirements for all of these instructions requires a working Retention Policy Services and Records Manager system.

6.9.1 Configuring the NiceLabel environment variable

By default, Records Manager is configured (after installation) to be used with NiceLabel 17 or later. To use the latest version of NiceLabel, make the configuration changes as follows:

1. After deploying `records.war`, on the Records Manager host, go to **System Properties > Environment Variables**.
2. To configure NiceLabel for Label Print operation, in the **Environment Variables** dialog box, in the **System Variables** area, click **New**.
3. In the **New System Variable** dialog box, in the **Variable name** box, type **LabelRenderPath**.
4. In the **Variable value** box, type the following value:

```
C:\<app_server>\records\prm\library\labelrender\NiceLabel10ConsoleApplication.exe
```

5. Click **OK**.

6.9.2 Setting up barcode label printing (rendering) and testing instructions

Follow these procedures after setting up Records Manager. The goal is to install Adobe Print Drivers, install NiceLabel, and optionally import your own templates. A procedure to test barcode label printing is also provided.

6.9.2.1 Installing Adobe Print Driver and NiceLabel

Label printing and rendering is dependent on both Adobe Acrobat Professional and NiceLabel, which are not included with the purchase of Records Manager.

To install and configure the Adobe Print Driver:

1. On the Records Manager machine, install Adobe PDF printer driver (available by installing Adobe Acrobat Professional).
2. Go to **Start > Control Panel > Printers** and locate the Adobe PDF printer.
3. Right-click the Adobe PDF printer and choose **Properties**.
4. On the **General** tab, click the **Printing Preferences** button.
5. Locate the **Adobe Output Folder** list box, click the **Browse** button and locate the **PDF_out** folder. By default, it is in `C:\Documentum\pdf_out\`.
6. Clear the following check boxes:
 - **View Adobe PDF results**
 - **Add document information**
 - **Delete log files for successful jobs**
7. Click **Apply** and then **OK**.
8. Install NiceLabel according to its installation guide and provide the license for the same in the NiceLabel Designer application. Make sure that it is working fine.



Note: After installing Adobe, make sure that you register the software, otherwise label printing will not work. OpenText recommends that you open a PDF after installing the software and if necessary fill in the registration information.

6.9.2.2 Testing barcode label printing

To test label printing (rendering):

1. Log in to the Records Client and navigate to the **Templates** cabinet.
2. Locate the template called Sample NiceLabel 17 Template. The template is available by default with on the Records Manager DAR. To use custom NiceLabel templates, refer to “[Optional steps to import custom NiceLabel Templates](#)” on page 62.
3. Right-click the template and select **Create Label Rule**.
4. Enter a name for the rule, select the object type you want to apply it to, select the attributes you want to account for on the printed label (keywords, object_name and subject for example) and then click **Finish**.

For example, type *rule for Physical Document* for the **Name**, *Physical Document (dmc_prm_physical_document)* for the object type it applies to, and select keywords, object_name, and subject for the attributes.

The default template has attributes for object name, keywords and subject. Customers, however, can create their own custom template with a full version of NiceLabel and add the attributes they want to have displayed on the printed label.

5. Add attributes (template contains subject, keywords, object_name attributes, at least one attribute should be added) and click **Finish**.
6. Verify that the new rule shows up in **Physical Records Manager > Label Printing Rules**.
7. Navigate to **Cabinets > System > Applications > PrmConfig > Prm DocbaseConfig** and set the filter to **Show All Objects and Versions**.
8. Right-click prm_docbase_config and select **Properties**.
9. Set the **Temporary Target Object Folder for Label Printing** on the **Info** tab (for example, `http://<hostname>:<port>/transformation/services and /TempLabelTarget`).



Note: The **Temporary Target Object Folder for Label Printing** is set to *Temp* by default. If you want to change it to something else, ensure you have created the Temp Label Target folder within the repository and that the users have Write permission on it.

10. Click **OK**.
11. Create a physical object type according to the rule created. On the **Info** tab, enter attributes specified in the rule (for example, subject and keywords). Then, on the **Physical Info** tab, enter a barcode.

The following screenshot of the physical document created against the rule shows the entries for the attributes selected in the rule.

The screenshot shows the 'Properties: Info' dialog box for a document named 'physDocument1'. The 'Info' tab is selected. The document type is 'dmc_prm_physical_document' and the format is blank. The 'Name' field contains 'physDocument1', the 'Title' field is blank, and the 'Subject' field contains 'paper for execution'. The 'Keywords' field shows 'paper1' with an 'Edit' link. At the bottom are 'OK' and 'Cancel' buttons.

Properties: Info

Info Physical Info Permissions History Transfer Info

physDocument1
Type: dmc_prm_physical_document
Format:

Name: physDocument1

Title:

Subject: paper for execution

Keywords: [Edit](#) paper1

OK Cancel

Figure 6-2: Properties of a physical document based on the rule specified

The next screenshot shows the value entered for the barcode.

The screenshot shows the 'Properties: Physical Info' dialog box for the same document. The 'Physical Info' tab is selected. The 'Barcode' field contains the value '071510150124'. Below it are four checkboxes: 'Lost', 'Marked for Export', 'Marked for Shipped', and 'Marked for Destruction', all of which are currently unchecked. At the bottom are 'OK' and 'Cancel' buttons.

Properties: Physical Info

Info Physical Info Permissions History Transfer Info

physDocument1
Type: dmc_prm_physical_document
Format:

Barcode: 071510150124

Lost: ☐

Marked for Export: ☐


Marked for Shipped: ☐

Marked for Destruction: ☐

OK Cancel

Figure 6-3: Barcode value

12. On the physical document created, right-click and select **Print Label** to generate a label from the resulting **Label Print Request** screen.
13. On the **Label Print Request** screen, click **OK**.

 **Note:** Sample NiceLabel 17 Template is applicable for NiceLabel 17 and NiceLabel 10.

PRTS processes the request when you click **OK** and then generates the label according to the rule specified. An initialization screen appears briefly which launches Acrobat and displays the barcode label.

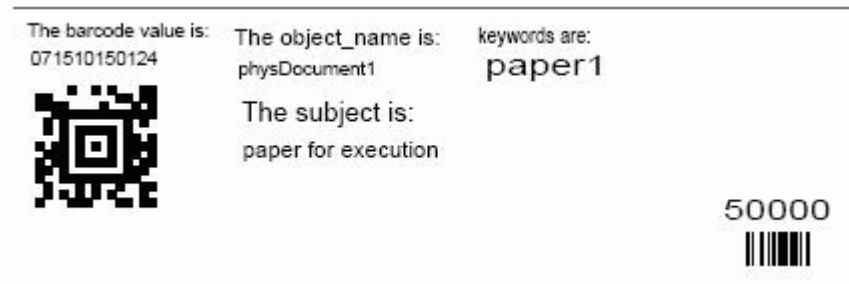



Figure 6-4: Printed label

6.9.2.3 Optional steps to import custom NiceLabel Templates

Although Sample NiceLabel 17.x template is provided, custom templates can be used for Label Printing. Follow these steps to import and set permissions for the template. When a new print label template is imported, the ACL needs to be changed to allow label administrators to save, modify, and destroy print label rules.

 **Note:** If the ACL is not changed, that is steps 8-12 were not completed, then only the person who imported the template could create, modify and/or delete a rule based on that template.

To import and set permissions for the template:

1. Using NiceLabel, create and save the template. Refer to NiceLabel documentation for help in creating the label.

 **Note:** Ensure that the selected Printer is set to Adobe PDF during the template creation.

2. Log in to **Records Client** and navigate to **Cabinets > Templates**.
3. Click **File > Import**.
4. On the **File Selection** dialog, click **Add Files** and navigate to the saved template. Select it and click **OK**.
5. Click **Next** on the **File Selector** screen.
6. On the **Object Definition** screen, make the following changes:
 - Change **Type** to **Label Template (dmc_prm_label_template)**.
 - Ensure **Format** is set to **NiceLabel 17 Template** for NiceLabel 17 and NiceLabel 10.

- Ensure **Template Provider** is set to **NiceLabel**.
- 7. Click **Finish**.
- 8. After importing the template, right-click the template and select **Properties**.
- 9. Click on the **Permissions** tab.
- 10. Click **Select** link for the **Permission Set Name**.
- 11. On the **Choose a permission set** dialog, select *dmc_prm_label_template_acl*, and then click **OK**.
- 12. Click **OK** to close the **Properties** dialog box.

6.10 Batch processing XML schema for Records Manager

This section is optional unless Records Manager is set up for batch processing operations.

6.10.1 Overview

The XML schema file described in this section is used to capture and upload barcode information scanned from a barcode scanner/reader regardless of its brand or level of sophistication.

Content in this section is intended to provide users an understanding of the various XML schemas used to maintain and keep the Records Manager inventory system up to date. This information is also intended for programmers who are creating applications used to generate batch processing files compliant to OpenText Documentum CM's physical records batch XML schema.

The scope of the information:

- Describes the batch processing operation for each XML schema to help you select the correct scanning operation on or from the barcode reader.
- Describes the XML tags, how they should be formatted for successful processing when a particular XML schema is uploaded.
- Describes how the barcode readings are transformed when an XML schema for a batch processing operation is uploaded.
- Provides examples of the XML schema formatting for each batch processing operation.

The XML schemas described are used to ensure barcode readings can be successfully uploaded and processed regardless of the barcode reader used and regardless of the format in which the scanned results are captured. XML is used as it is well suited for transmitting complex data structures of any type. The output results of the scanner, regardless of the format or encryption technique, are basically transformed to the

appropriate XML file for the desired operation described in “[Batch processing operations](#)” on page 69.

The format of the XML file is of primary interest, not the barcode reader nor the format of its output. It does not matter what format the scanned readings are in, as long as the XML file is properly formatted according to the examples provided in this section.

A separate XML file can be created for each operation or a combination of operations can be included in one XML file.

After the scanning is completed, the user needs to upload the XML file using the Records Client or Auto-Queue. Processing will start after the upload is complete.

The following table identifies the optional parameters that can be included in each of the batch processing operations.

Table 6-5: Optional parameters usage

Operation	Additional Info	Created At	Location	Sent to	Use Transaction	Recursive
Reconciliation	x	x				
Mark as Destroyed	x	x			x	
Mark as Shipped	x	x		x	x	x
Mark as Exported	x	x		x	x	
Charge-in	x	x	x			x
Move to Container	x	x	x		x	

The optional parameters are described as follows:

Table 6-6: Optional parameter descriptions

Optional Parameter	Description
Additional Info (additional_info)	This optional parameter can be included for any remarks when necessary. After the operation is processed, this information is added to the report. This information is not processed by the operation.

Optional Parameter	Description
Created At (created_at)	<p>A timestamp can be specified which can be used to prevent two different batch operations from overwriting each other if they were not processed in the correct order.</p> <p>WARNING If the system determines that the batch operation is out of date, the operation will have a status of ok but nothing will be done (the operation is skipped). This is useful if someone scans a set of objects but does not upload until later. If someone else updates the status of the same objects, you do not want the old (out-of-date) information processed.</p>
Location (location)	The location represents the physical container that the operation will occur on. The physical item will be linked into this folder.
Sent To (sent_to)	The name of a Retention Policy Services address object where the object was sent to. The physical item will not be linked into the address. Instead the next location will be set if this is done on a mark as shipped operation.
Use Transaction (use_transaction)	Indicates whether a transaction should be used for the operation or not. If set to yes, and any of the actions fail, the system will rollback so that no action is done to any of the items in the list (as if the actions were not attempted). By default, this is false, set to no, which means the system will not use a transaction. Rollbacks will be done on a best effort basis.
Recursive (recursive)	If the item(s) scanned are physical containers, perform the operation of the contents in the container recursively. This parameter is set to no by default and therefore all items must be scanned. For example, items B and C in container A must be scanned along with container A. If set to yes only item A needs to be scanned; items B and C are automatically processed.

The following XML file contains the XML schema definitions (XSD) of the elements for all of the batch processing operations described in [“Batch processing operations” on page 69](#). These definitions apply to all of the batch processing examples in that chapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="action_list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:group ref="action_set" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute ref="notify" default="none" />
      <xsd:attribute name="user" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
```

```

<xsd:complexType name="barcode_list_type">
<xsd:sequence>
<xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
<xsd:attribute ref="use_transaction" use="optional"/>
</xsd:complexType>
<xsd:element name="barcode_list">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="barcode" minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="barcode" type="barcode_type"/>
<xsd:complexType name="barcode_type">
<xsd:simpleContent>
<xsd:extension base="barcode_base">
<xsd:attribute ref="input" use="optional" default="scanned"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="barcode_base">
<xsd:restriction base="xsd:string">
<xsd:pattern value=".+"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:element name="sent_to" type="xsd:string"/>
<xsd:attribute name="input">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="scanned"/>
<xsd:enumeration value="keyed"/>
<xsd:enumeration value="other"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="use_transaction" type="yes_no_type" default="no"/>
<xsd:attribute name="recursive" type="yes_no_type" default="no"/>
<xsd:attribute name="created_at" type="xsd:dateTime"/>
<xsd:attribute name="notify">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="none"/>
<xsd:enumeration value="on_error"/>
<xsd:enumeration value="always"/>
<xsd:enumeration value="on_completion"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:group name="action_set">
<xsd:choice>
<!-- ADD OPERATIONS -->
</xsd:choice>
</xsd:group>
<xsd:simpleType name="yes_no_type">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="yes"/>
<xsd:enumeration value="no"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:element name="additional_info" type="xsd:anyType"/>
<!-- ADD DEFINITIONS -->
</xsd:schema>

```

The following list describes the tags in the XML file, and the order they need to be in, for all batch processing operations.

- The first tag, `<?xml version="1.0" encoding="UTF-8"?>`, is a descriptive tag at the top of the XML file that identifies the file as an XML file along with the version and encoding.
- The second tag, `<?dctm xml_app="ignore"?>`, is a processing instruction that must be included to prevent the XML file from being parsed by Documentum when importing the file.

The information has to be uploaded intact.

- The third tag, the `<action_list>` identifies the desired batch processing operation and the objects that need to be scanned.



Note: The barcode items listed for a batch processing operation within the action list should be in the proper order though it does not matter which items are actually scanned/swiped first. For example, the barcode for a container item should be the first entry followed by its contents as indicated in the following batch processing operation. A processing error will occur if the `<barcode_list>` is put before the `<container>`. The system knows that you cannot put a physical container into a physical document.

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list notify="always">
  <reconciliation>
    <container>conBarcode1</container>
    <barcode_list>
      <barcode>physDocBarcode1</barcode>
      <barcode>physDocBarcode2</barcode>
    </barcode_list>
  </reconciliation>
</action_list>
```

Optional parameters can be added to the action list so that the respective information can be analyzed when you view the report after a batch processing operation is uploaded, that is when you select **View Report** from the **Operation Status** screen. Optional parameters within the action list for a particular operation can be included as indicated in the preceding Optional parameters table.

The following list identifies the optional attribute names and describes the choice of value options, if any, that can be defined for each:

- *input*: barcode entries are identified on the report as one of the following:

Input values are meant as a guide (especially when determining when a barcode value could not be found) and there is no special business logic associated.

- *scanned*

This entry for the input indicates that the barcode readings were scanned. In this case, the user could not make a mistake (the barcode is not in the system).

- *keyed*

This entry for the input indicates that the barcode readings were manually keyed in using the keypad of the barcode reader. In this case, it is possible since the value was keyed in that a typo was made.

- *other*

The barcode value was neither scanned nor keyed. The value was determined by other means.

- *use transaction*: can be set to *yes* or *no*, and is set to *no* by default.

Processing can be aborted and rolled back if any operation fails when set to *yes*.

- *recursive*: can be set to *yes* or *no*, and is set to *no* by default.

- If set to *yes*, and if the item(s) scanned are physical containers, system will perform the operation of the contents in the container recursively.

- If set to *no*, which is the default, all items must be scanned. For example, if set to *no* items B and C in container A must be scanned along with container A. If set to *yes* only item A needs to be scanned; items B and C are automatically processed.

- *created at*: provides a timestamp if necessary, which is used to prevent batch processing when two people perform the same scanning operation against the same objects but upload at different times. The system can compare timestamps and prevent processing of the older instance.

- *notify*: the value options for this attribute can be set to one of the following:

- *none*

No notification is sent to anyone. This is the default value if not specified.

- *on_error*

Notifications will be sent only when an error occurs.

- *always*

Notifications will always be sent.

- *on_completion*

Notifications will be sent only when the operation is complete.

- *user*:

- If a user is specified, the notifications will be sent to their OpenText Documentum CM Inbox. This value needs to be the value used to log in to the OpenText Documentum CM system (known as the user Login name).



Note: 1. When setting up an account in OpenText Documentum CM, there is a user name field and a user Login name field which do not need to be the same. This application is looking for the user Login name.

2. The specified user must also be in the role `dmc_prm_batch_upload`. Users in this role will have the capabilities of a library administrator and an inventory manager.

- If a user is not specified:
 - If manual upload, notifications will be sent to the logged in user
 - If asynchronous processing, notifications will be sent to the install owner



Note: If a media plug-in error occurs, the install owner will get a notification.

6.10.2 Batch processing operations

All elements that are used to create the XML file for each of the batch processing operations are defined in the XSD file described in [“Overview” on page 63](#).

6.10.2.1 Overview of operation status and view report

The status and the details of a batch operation , whether it was processed successfully or not can be determined from the **Operation Status** tab, when you right-click the XML file and select **Properties**. The **Status**, whether processing *Failed* or *Executed Ok*, is displayed directly on the Operation Status tab, using `mark_as_exported` as an example.

The details against each status result displayed, can be determined when you click **View Report**.

[Figure 6-5](#) shows details in Operation#1 against the *Failed* status. Details in the report can then be used to identify and correct the problem. [Figure 6-6](#), according to Operation#2 against the *Executed Ok* status, shows new details after the problem is corrected and a second attempt is made.



Figure 6-5: Failed status details

All possible failure messages in the report displayed against any of the batch operations, are listed in the respective batch operations in the following sections.



Figure 6-6: Executed Ok status details

Details in the report are itemized line by line against all batch operations, as shown in [Figure 6-5](#) and [Figure 6-6](#), except for Reconciliation operations as shown in [Figure 6-7](#). Each line item in the report of each batch operation, except Reconciliation, is described in the following tables:

Table 6-7: Line items in export reports

Mark As Exported	
Line Item	Description
1.	Identifies the batch operation.
2. Sent to:	If displayed, identifies the name of the export address.
3. Created at:	Identifies the time of the upload.
4. Started at:	Identifies the time Records Manager processed the XML file.
5.	Repeating. Identifies the barcode value the second item identifies the name of the object scanned.
6.	Displays the status. Also, includes the error message if the upload fails.
7. Finished at:	Identifies the time Records Manager finished processing the XML file.

Table 6-8: Line items in charge-in reports

Charge-In	
Line Item	Description
1.	Identifies the batch operation.
2. Created at:	Identifies the time of the upload.
3. Started at:	Identifies the time Records Manager processed the XML file.
4. Charged-in item:	Identifies the object name and its barcode value and the location.
5. Finished at:	Identifies the time Records Manager finished processing the XML file.

Table 6-9: Line items in mark as destroyed reports

Mark As Destroyed	
Line Item	Description
1.	Identifies the batch operation.
2. Created at:	Identifies the time of the upload.

Mark As Destroyed	
Line Item	Description
3. Started at:	Identifies the time Records Manager processed the XML file.
4.	Identifies the object name and the barcode value of the object that was destroyed.
5.	Identifies the status, whether Records Manager processing completed ok or not.
6. Finished at:	Identifies the time Records Manager finished processing the XML file.

Table 6-10: Line items in mark as shipped reports

Mark As Shipped	
Line Item	Description
1.	Identifies the batch operation.
2. Created at:	Identifies the time of the upload.
3. Started at:	Identifies the time Records Manager processed the XML file.
4.	Identifies the object name and the barcode value of the object that was shipped and if it got shipped successfully.
5.	Identifies the status, whether Records Manager processing completed ok or not.
6. Finished at:	Identifies the time Records Manager finished processing the XML file.

Table 6-11: Line items in move to container reports

Move To Container	
Line Item	Description
1.	Identifies the object name and the barcode value of the object that was moved and the name of the container and barcode value to which it was moved.

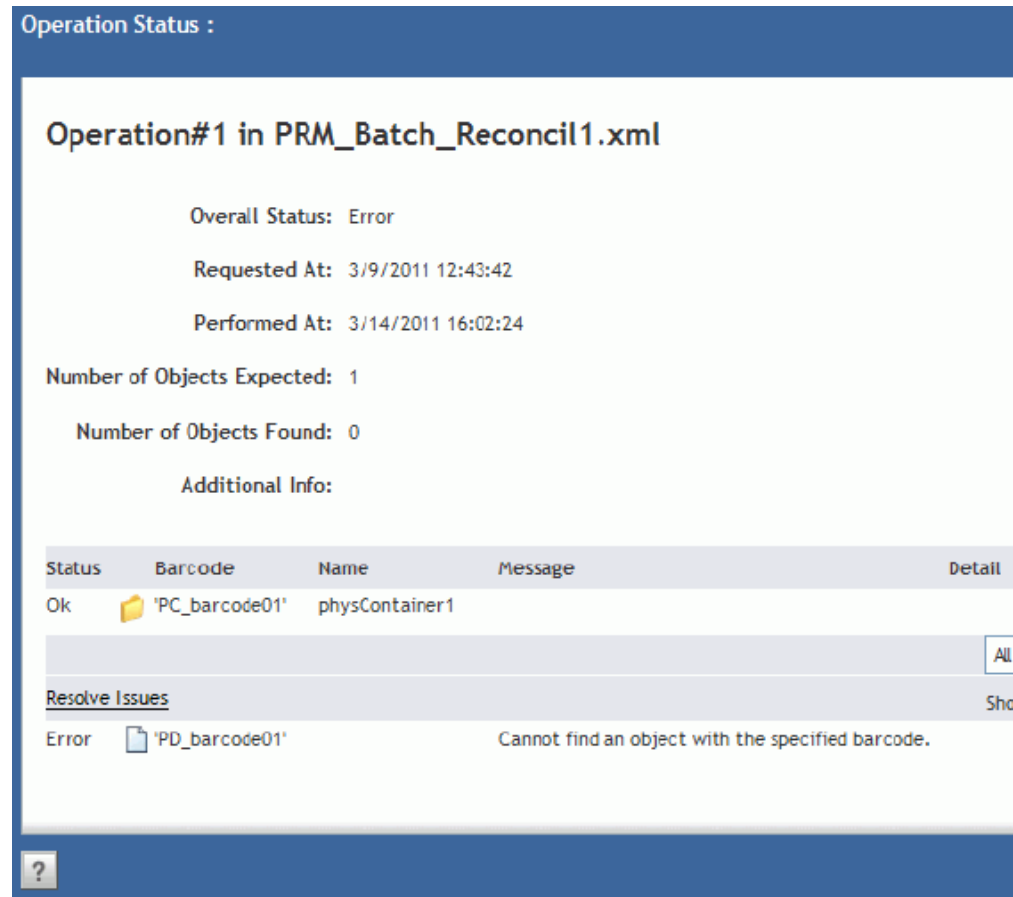


Figure 6-7: Only reconciliation details are formatted differently

Although any issues encountered during the processing of Reconciliation operations are listed under Resolve Issues, only those that can be resolved will be resolved automatically when you click Resolve Issues. For the list of issues that can be resolved, refer to [“Resolvable issues” on page 76](#).

Each line item in Reconciliation reports is described as follows:

Table 6-12: Line items in reconciliation reports

Reconciliation	
Line Item	Description
1. Overall Status:	Indicates whether the upload executed successfully or not.
2. Requested at:	Identifies the time of the upload.
3. Performed at:	Identifies the time Records Manager processed the XML file.

Reconciliation	
Line Item	Description
4. Number of Objects Expected:	Number of items expected within the container.
5. Number of Objects Found:	Actual number of objects found within the container.
6. Additional Info:	Allows additional information to be printed on the report (written in the XML file).
7. Status, Barcode, Name, Message, Detail	All objects scanned and processed successfully are listed here. The example in Figure 6-7 , indicates that the barcode matched the name of the object and was scanned successfully.
8. Resolve Issues	Objects that could not be processed successfully are listed under this item. If you click Resolve Issues, only those issues listed here that can be resolved would be resolved automatically. For a list of resolvable issues, refer to “Resolvable issues” on page 76 .

6.10.2.2 Reconciliation

Use this operation to reconcile the contents of one or more physical container objects. This operation compares the scanned physical container and its contents to what the system has recorded and reports any discrepancies. From this report, it is possible to resolve discrepancies. All issues that can be resolved are listed in [“Resolvable issues” on page 76](#).

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

Example 6-1: SimpleReconcile.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list notify="always">
  <reconciliation>
    <container>123456789101</container>
    <barcode_list>
      <barcode>234567893467</barcode>
      <barcode>234567893468</barcode>
    </barcode_list>
  </reconciliation>
</action_list>
```



Elements of the reconciliation operation shown previously are defined in the following XML schema definitions.

```

<xsd:element name="reconciliation"/>
<xsd:complexType name="reconciliation_type">
  <xsd:sequence>
    <xsd:element ref="additional_info" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element name="container" minOccurs="0" maxOccurs="1"
      type="barcode_type"/>
    <xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute ref="created_at" use="optional"/>
</xsd:complexType>

```

The following table lists the possible notification messages that can be sent to the OpenText Documentum CM Inbox when the scanned results for a reconciliation operation are uploaded and processed by Records Manager. Notification messages are sent to the user defined in the XML file, or if not defined, to the user who uploaded the file. However, the notify tag needs to be used and set accordingly (refer to [“Overview” on page 63](#) for an explanation of the different values that can be set for it). Next to the notify tag, you can state the user with the tag: user=(userLoginName). Both tags are within the action_list parameter.

Table 6-13: Notification messages for reconciliation operations

Notification Message	Notes
Object scanned found in system but is linked into a different container	The scanned physical object is linked into a non-physical container object instead of the physical container in which it was scanned. The scanned object for example, might have been inadvertently linked into a regular folder such as My Home Cabinet.
Object scanned found but found in a different container than expected	The scanned object was found in a physical container.
The barcode for the scanned container is not a folder	The scanned object identified is neither a container nor a physical container.
Scanned object not found in expected location	The scanned object was found at a location other than the one reported.
System indicates that this object should be in container but was not scanned	There are more physical objects inside the scanned container than reported by the scan.
Scanned object was found but the system indicates that object is not available	An item is unavailable if it: <ul style="list-style-type: none"> • is marked for destruction • is marked for export • is marked as destroyed • is marked as shipped for export • is marked as lost
System could not find any object with specified barcode	This error may be reported with an existing object, but the user does not have browse permissions to the item.

Notification Message	Notes
The system has found more than one object that uses the same barcode. Please check with administrator	
Scanned object has different home location than the container's	
Scanned object has different current location than the container's	
Unexpected Error during processing	
Scanned object is marked as Charged-out	

6.10.2.2.1 Resolvable issues

There are currently two reported issues that can be resolved from the reconciliation report page, that is the report the user can view from the Operation Status tab on the Properties of the XML file:

- Scanned object is marked as Charged-out
If the user decides to resolve this issue, the object would be charged in.
- Object scanned found but found in a different physical container than expected.
The physical address attributes will not be changed.
If the user decides to resolve this issue, the object would be moved to the physical container that is identified by its barcode on the batch operation XML file.
The system will guard against trying to move a container inside itself.

6.10.2.3 Charge-in

Use this operation when one or more physical objects, that were taken from the inventory system, are returned to the inventory system. Physical objects that have been charged out, need to be scanned back into the Records Manager inventory system when they are returned/charged in.

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

Example 6-2: charge_in.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list notify="always" user="radmin">
  <charge_in recursive="no" created_at="2008-07-22T14:52:40">
    <additional_info>
      This space can be used optionally to add a
      note for example and can otherwise remain empty
    </additional_info>
  </charge_in>
</action_list>
```

```

        <location>Bin1</location>
        <barcode_list>
          <barcode>87652381</barcode>
        </barcode_list>
      </charge_in>
    </action_list>

```



In the preceding example, assuming the object has not been charged-in or moved since July 22, 2008:

- The object that has barcode 87652381 will be charged-in
- The object that has barcode 87652381 will be moved into location with barcode value *Bin1*



Note: If location *Bin1* (which represents the barcode of a physical container) does not exist, or is marked lost, none of the items in the list will be processed. This means the items will not be charged in.

Elements of the charge-in operation shown previously are defined in the following XML schema definitions.

```

<xsd:element name="charge_in"/>
<xsd:complexType name="charge_in_type">
  <xsd:sequence>
    <xsd:element ref="additional_info" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="location" minOccurs="0" maxOccurs="1" type="
      barcode_type"/>
    <xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute ref="recursive" use="optional"/>
  <xsd:attribute ref="created_at" use="optional"/>
</xsd:complexType>

```

The following table lists the possible notification messages that can be sent to the OpenText Documentum CM Inbox when the scanned results for a charge-in operation is uploaded and processed by Records Manager.

Table 6-14: Notification messages for charge-in operations

Notification Message	Notes
Was not charged-out at the time that this operation was requested	Only a charged out object can be charged in.
Has been charged-in since this operation was requested. The operation has been skipped	
Has been charged-out since this operation was requested. The operation has been skipped	If the state of the object changes after the scan, as reported under the optional parameter created at, the operation is skipped.

6.10.2.4 Mark as destroyed

Use this operation when one or more physical objects are destroyed. Only physical objects that were marked for destruction can be marked as destroyed. There is no batch operation for marking physical objects for destruction. Marking for destruction is done automatically for physical objects under retention when disposition is run. Physical items that are not under retention can be marked for destruction using the Records Client.

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

Example 6-3: markasdestroy.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list user="batchupload_user">
  <mark_as_destroyed>
    <additional_info>
      testing markAsDestroyed
    </additional_info>
    <barcode_list>
      <barcode>678584784746</barcode>
    </barcode_list>
  </mark_as_destroyed>
</action_list>
```



Elements of the mark as destroyed operation shown previously are defined in the following XML schema definitions.

```
<xsd:element name="marked_as_destroyed"/>
<xsd:complexType name="mark_as_destroyed_type">
  <xsd:sequence>
    <xsd:element ref="additional_info" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute ref="use_transaction" use="optional"/>
  <xsd:attribute ref="created_at" use="optional"/>
</xsd:complexType>
```

The following table lists the possible notification messages that can be sent to the OpenText Documentum CM Inbox when the scanned results for a mark as destroyed operation is uploaded and processed by Records Manager.

Table 6-15: Notification messages for mark as destroyed operations

Notification Message	Notes
Has been destroyed or charged-out since this operation was requested. The operation has been skipped	

6.10.2.5 Mark as exported

Use this operation when one or more physical objects are transferred from one inventory system to another, that is from one home location address to another. Physical objects that are exported, that is when the logical ownership of its real-world object is transferred to another address for pick up, from the current inventory system to another inventory system must be identified, scanned and marked as exported. As an example, the physical object requested needs to be transferred from one warehouse to another warehouse for pick up.



Note: This operation should not be confused with the Move to Container operation used to scan barcodes of physical objects that were in a particular container object but are now located in a different physical container object within the same inventory system.

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.



Example 6-4: markasexported.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list>
  <mark_as_exported>
    <additional_info>
      testing markAsExported
    </additional_info>
    <sent_to>Documentum</sent_to>
    <barcode_list>
      <barcode>763647528469</barcode>
    </barcode_list>
  </mark_as_exported>
</action_list>
```



Elements of the mark as exported operation shown previously are defined in the following XML schema definitions.

```
<xsd:element name="marked_as_exported"/>
<xsd:complexType name="mark_as_exported_type">
  <xsd:sequence>
    <xsd:element ref="additional_info" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element ref="sent_to" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute ref="use_transaction" use="optional"/>
  <xsd:attribute ref="created_at" use="optional"/>
</xsd:complexType>
```

No unique messages are sent as part of the notification for the mark as export operation.



Note: All generic error messages may still be reported. For example, a message indicating that the barcode was not found.

6.10.2.6 Mark as shipped

Use this operation, against charged out objects, when one or more physical objects that have been shipped, reach their destination.

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

- Mark as shipped only works for objects that need to be shipped because of a charge-out operation.
 - Mark as shipped will fail if an export was initiated.
- If the sent_to attribute does not match where the system thought the physical object should go, an attribute on the physical object (aspect attribute) will be set. The value can be viewed from the Physical Info tab when you select the properties of the physical object.

Example 6-5: markasshipped.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list>
  <mark_as_shipped>
    <additional_info>
      testing markAsShipped
    </additional_info>
    <sent_to>Warehouse19</sent_to>
    <barcode_list>
      <barcode>785937249568</barcode>
    </barcode_list>
  </mark_as_shipped>
</action_list>
```



Elements of the mark as shipped operation shown previously are defined in the following XML schema definitions.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

  <xsd:element name="marked_as_shipped"/>
  <xsd:complexType name="mark_as_shipped_type">
    <xsd:sequence>
      <xsd:element ref="additional_info" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="sent_to" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="barcode_list" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute ref="use_transaction" use="optional"/>
    <xsd:attribute ref="recursive" use="optional"/>
    <xsd:attribute ref="created_at" use="optional"/>
  </xsd:complexType>

</xsd:schema>
```

The following table lists the possible notification messages that can be sent to the OpenText Documentum CM Inbox when the scanned results for a mark as shipped operation is uploaded and processed by Records Manager.

Table 6-16: Notification messages for mark as shipped operations

Notification Message	Notes
The charge-out state of Object has been changed since this operation was requested. The operation has been skipped	There should not be a change in the state of the item after the time of the scan.

6.10.2.7 Move to container

Use this operation when one or more physical objects are moved from one physical container object to another physical container object within the same inventory system. A physical object may have been moved to another container or could have been charged out.

To determine or to view the operation status or the report of any batch processing operation, refer to the following section To view the operation status and/or to view the report for an operation in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

Example 6-6: move_at_specific_time_example.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?dctm xml_app="ignore"?>
<action_list>
  <!-- move it back to its original container -->
  <move_to_container created_at="2008-06-13T09:31:45">
    <location>Bay5</location>
    <barcode_list>
      <barcode>356465757682</barcode>
    </barcode_list>
  </move_to_container>
</action_list>
```



Elements of the move to container operation shown previously are defined in the following XML schema definitions.

```
<xsd:element name="move_to_container"/>
<xsd:complexType name="move_to_container_type">
  <xsd:sequence>
    <xsd:element ref="additional_info" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element name="location" minOccurs="0" maxOccurs="1"
      type="barcode_type"/>
    <xsd:element ref="barcode_list" minOccurs="1"
      maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute ref="use_transaction" use="optional"/>
  <xsd:attribute ref="created_at" use="optional"/>
</xsd:complexType>
```

The following table lists the possible notification messages that can be sent to the OpenText Documentum CM Inbox when the scanned results for a move to container operation is uploaded and processed by Records Manager.

Table 6-17: Notification messages for move to container operations

Notification Message	Notes
Object was charged-out at the time of this request. It has been charged-in	In this case, the move operation will charge the item in.
Object has been moved or charged-out since this operation was requested	Move operation will skip the item.
Object was removed from the scanned container	This message is printed for information only to inform the unlinking of the item from the incorrect container.
Object is already present in the scanned container	Move operation will skip the item.
Object moved to the container specified	Move operation completed successfully.
An error is displayed if the barcode scanned for the physical container is not found in the system	

6.11 Updating the pass-along workflow

Refer to these instructions to ensure the pass-along workflow forwards the task successfully.

To update the pass-along workflow:

1. Launch Workflow Manager. Workflow Manager comes with the full composer product.
 - a. Double-click `launch_wfm.bat` to start Workflow Manager.
 - b. Enter the repository information to connect (log in as the install owner).
2. Click **File > Open** and double-click the passalong workflow in the repository located in `/System/Applications/prm/PassalongWorkflow`.
3. Uninstall the template from **File > Uninstall**.
4. Open the **NotifyRecipientItemsSent** activity with the **Activity Inspector** as follows:
 - a. Select the **NotifyRecipientItemsSent** activity.
 - b. Double-click the activity to launch the **Activity Inspector** or use the **Tools > Activity Inspector** menu.
5. Select the **Transition** tab and modify the settings as follows:
 - a. Change the option selected for **How are next activities selected when activity completes?** in the list box from **Select all connected activities** to **Let the activity's performer choose**. Enter the values for this option according to the remaining steps.

- b. Set **Select up to # next activities** to **Select up to 1 next activities**.
 - c. Click **Start only forward activities**.
 - d. Click **Apply** to accept changes and then click **OK** to finish editing.
6. On the Workflow Manager screen click: **File > Save** to save the changes, then **File > Validate** to validate the workflow, and lastly **File > Install** to install the modified workflow.

6.12 Configuring the disposition job to run successfully against objects with an Unknown disposition strategy

Configuration of the disposition job for the Unknown disposition strategy is no longer necessary since this is now configurable on the Retention Policy Services Disposition Configuration file, as instructed in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)*.

6.13 Configuring the parameters for the Retention Policy Services notification job

The Retention Policy Services jobs and method arguments descriptions table in the *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* contains the information for configuring the parameters.

6.14 Setting up Records Queue Manager

Follow the instructions in this section to set up Records Queue Manager.

Records Queue Manager is a queueing service that is a packaging of Transformation Services, with only the profiles required for remote processing. Records Queue Manager is used in conjunction with the Work Order Framework (WOF) for processing work orders remotely and asynchronously. With RQM, scalability is provided for the Records products. Additional Records Queue Manager servers can easily be set up to handle more workloads. With Records Queue Manager, the processor-intensive requirements have been shifted from the application server to any number of dedicated Records Queue Manager servers. Refer to the product *Release Notes* available on My Support (<https://support.opentext.com>) to determine server requirements for Records Queue Manager.

Download the following files:

- Records Queue Manager tar file
- Transformation DAR
- Rich Media Services DAR



Note: Make sure that Records Queue Manager is not installed in the same server as OpenText Documentum Content Management (CM) Transformation Services.

The following packages are required to be installed before installing Records Queue Manager on Red Hat, 64-bit:

- glibc-2.12-1.80.el6.i686.rpm
- nss-softoken-freebl-3.12.0-11.el6.rpm
- libgcc-4.4.6-4.el6.i686.rpm
- libXau-1.0.5-1.el6.i686.rpm
- libxcb-1.5-1.el6.i686.rpm
- libX11-1.3-2.el6.i686.rpm
- libXext-1.1-3.el6.i686.rpm
- libXi-1.3-3.el6.i686.rpm
- libXtst-1.0.99.2-3.el6.i686.rpm

Prerequisite:

- End users must have JDK pre-installed. The JAVA_HOME and PATH environment variables must be set before installing and setting up Records Queue Manager.
- Make sure that the Microsoft Visual C++ 2022 x64 Redistributable package is installed on the server host.

To set up Records Queue Manager on Windows and Linux:

1. Grant the install owner logon as service rights according to the following steps:
 - a. Click **Start > Settings > Control Panel**.
 - b. Double-click **Administrative Tools**.
 - c. Double-click **Local Security Policy**.
 - d. Open **Local Policies**.
 - e. Open **User Rights Assignment**.
 - f. Open **Log On as a Service**.
 - g. Click **Add**.
 - h. Select the user you want to grant logon service access to and click **OK**.
 - i. Click **OK** to save the updated policy.
2. Extract the contents of the Records Queue Manager tar file and run the Setup executable file (rqmSetup.exe on Windows or rqmSetup.bin on Linux).
Setup will install the following products and support components:

- Records Queue Manager
 - Transformation Services
 - DFC Runtime Environment
3. Click **Next**.
 4. Select **I accept the terms of the license agreement**, and then click **Next**. The default destination directory (C:\Documentum) for the installation of all components (Transformation Services Framework, Records Queue Manager, and Foundation Java API) appears on the screen.
 5. Change the destination directory, if required, and then click **Next**. To accept the default, click **Next**.



Note: Ensure that the selected destination directory does not contain a space in its name. Foundation Java API is installed in the same location as Transformation Services.

6. Click **Done** once the installation finishes.
7. Copy the `Transformation.dar` and `Rich_Media_Services.dar` files, and move them to the Documentum CM Server.
8. On the Documentum CM Server, use Documentum Composer to install these two DAR files. The order in which these DAR files are installed does not matter. These DARs must be installed on the global repository and on any repository you plan to work on.



Note: Ensure that you do not install any lower version of DAR files if you already have the higher version of the DAR files installed in your environment.

9. To complete the Records Queue Manager installation, configure the instance to a repository.



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

To configure an instance of Records Queue Manager server:

1. On the Records Queue Manager host, run the **RQM Configuration Utility**, from **Start > Programs > RQM Configuration Utility**, against all the repositories you installed the two DAR files on, starting with the Global repository.

On Linux, navigate to `$DOCUMENTUM/CTS/server_install` and then launch `rqmConfigurator.sh` to run the **RQM Configuration Utility**.
2. If you want to configure Vault secrets, on the **Documentum Server RQM Vault Components** page, select the **Enable Vault** check box.
3. If you selected the **Enable Vault** check box, provide the following information and click **Next**:

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

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

- b. **DSIS Token:** Provide the `dsis.dctm.token` token value as described in [“Configuring Documentum Secret Integration Service” on page 14](#).

! Important

If you enabled the Vault configuration, the configurator retrieves all the password information automatically from Vault. This includes the `rqmSysAdminPass`, `DFC_GLOBALREGISTRY_PASSWORD`, and `INSTALL_OWNER_PASSWORD` passwords.

4. Enter the host name of the machine that will be running the Records Queue Manager Framework. Correct the host name if it is invalid or append the domain to create a Fully-Qualified Domain Name. Click **Next**.
5. Enter the port number to which the Content Transformation Services Administration Agent will be configured. This port is used in conjunction with Documentum Administrator to communicate with the Records Queue Manager host. The default port number is 9095. Modify the default port number, if required, and click **Next**.
6. Enter the port number to which the Records Queue Manager services will be configured. The default port number is 9096. Click **Next**.
7. Enter the host name and port number of the primary connection broker. The default port number is 1489.
8. Select **Use certificates**. Click **Browse** to locate a valid `dfc.keystore` file. Enter the password for the located `dfc.keystore` file.



Note: This step is optional and is performed only when you want to configure this Records Queue Manager instance to a repository that is projected with a secured connection broker (certificate-based). Alternatively, you can select **Use Default Java TrustStore** while configuring Records Queue Manager to a secured Documentum CM Server.

9. Enter the global repository name, login name, and its password and click **Next**.



Note: This screen appears only when you are configuring a Records Queue Manager for the first time on this machine.

10. Select **Add a Records Queue Manager instance to a repository** and click **Next**.



Note: If a RQM product has already been configured in this repository, an additional option, **Remove a Records Queue Manager instance from a repository** appears along with the **Add a Records Queue Manager instance to a repository** option.

11. The Configurator detects the name of the RQM Server host. Correct the host name if it is invalid or append the domain to create a Fully Qualified Domain Name.
12. Click **Next**. The Repository Name and User Information screen appears.
13. Enter following information for the repository.
 - a. Select a repository to be served by the Records Queue Manager Server.
 - b. Enter the SuperUser name that Transformation Services will use to access the repository.
 - c. Enter the password for the SuperUser account.
 - d. Enter the domain name, if applicable. This is an optional field; if you are installing from a local machine, this can be left blank.
 - e. Enter the name of the repository user who will receive notifications from Records Queue Manager Server, otherwise known as the system operator.
 - f. Select the **Add Additional Domain Users** option, if you are setting up to support requests from additional domains. If you have multiple domain users in Documentum CM Server that transform content, select this option. The installer will create an Admin user for each domain to support transformations by users of each domain. Enter credentials for a new domain and its SuperUser in the required fields. Repeat this step to include additional domains, as required.
14. Click **Next**. If you cannot continue, check with the repository administrator to verify that the users exist and that you entered the correct password.
15. Click **Next**. The User Authentication screen appears.
16. Enter the system administrator user name and password. Click **Next**. Click **Done**. The Configurator closes.
17. After the configuration is complete, verify that the Records Queue Manager service from the Windows services is started.

On Linux, you will have to start the Records Queue Manager services manually. From the terminal change directory to \$DOCUMENTUM/CTS/server_install/CTSService and run the startRQM.sh, for example, `> ./startRQM.sh start`. While installing Records Queue Manager in Linux, set environment variables for DOCUMENTUM and DOCUMENTUM_SHARED.

18. Optionally, to verify only on Windows, run the following DQL statement against the repository to verify the `cts_instance_info`:

```
Select websrv_url, hostname from cts_instance_info
```

It should return the websrv_url and list `http://[host]:9096/cts/`.

19. Optionally, run this query to verify that the Records Queue Manager profiles were properly deployed to the repository:

```
select r_object_id, object_name, filter_names, filter_values from dm_media_profile
```

You should see results returned as in the following example:

Table 6-18: Results returned

r_object_id	object_name	filter_names	filter_values
083e92c480003aba	workorder_process	Visibility,CTSProdu ct	System,RQM

20. Proceed to [“Setting up Documentum Administrator”](#) on page 48.



Note: To register this instance for Privileged DFC, refer to [“Registering all Foundation Java API clients”](#) on page 48 for instructions.

21. Restart the RQM service.



Notes

- Make sure that you clear the BOF cache from the Web application server and Records Queue Manager environment after upgrading the Transformation Services DAR files and restart all the services.
- If you want to enable the generation of the disposition status in the form of logs on the Records Queue Manager server, then change the DfLogger statements (logging level) to DEBUG mode in the log4j2.properties file.

6.14.1 Installing and configuring Records Queue Manager in silent mode

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

1. Log in to the Records Queue Manager machine as an administrator.
2. Ensure that no programs or applications are running on the host.
3. Navigate to the folder containing the extracted Records Queue Manager installation files.
4. Locate the `rqm_install.properties` file.
5. Mention the installation path as a value for `CTS.INSTALLATION_DIR`.

```
## silent install response file

INSTALLER_UI=silent

####installation
CTS.INSTALLATION_DIR=<Please enter the installation path>
```

6. Open the command prompt and navigate to the installer directory where the `rqm_install.properties` file is available.
7. For Windows, run the installer silently from the command prompt, using this command:

```
rqmSetup.exe -f rqm_install.properties
```

For Linux, run the following command:

```
rqmSetup.bin -f rqm_install.properties
```

8. Verify the installation logs created under the Logs folder in the installer directory.
9. Upon successful installation, navigate to %RQM_HOME%\server_install\server_install folder. This folder must contain the following files: rqm_config_adding.properties and rqm_config_removing.properties.



Note: The %RQM_HOME% directory refers to C:\Documentum\CTS unless changed during the installation process.

10. To configure Records Queue Manager to a repository, update the rqm_config_adding.properties file with the relevant values using a text editor:

```
#all

#Specify the Admin port and Jetty port at first time config
CTS.ADMIN_PORT=9095
CTS.JETTY_PORT=9096
# Please specify the config actions, "add" or "remove".
CTS.INSTALL_TYPE=add
# Please specify the host name for CTS machine
CTS.HOSTNAME=<please enter the valid HOSTNAME>
# Please specify the repository name
CTS.DOCBASE=repo1
# Please specify the repository super user and password
CTS.DOCBASE_SUPERUSER=Administrator
CTS.NOT_ENCRYPTED_PASSWORD=password
# Please specify the domain name (optional)
CTS.DOMAIN=
# Please specify the user who receive notifications from CTS
CTS.DOCBASE_USER=Administrator
# Please set value to "true" if adding additional domain users, otherwise set it to
"false"
CTS.ADDITIONAL_DOMAINS=false
# Please set value to "true" if selecting this repository as Logging Performance
repository, otherwise set it to "false"
CTS.IS_PERFORMANCE_LOG_REPO=false

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

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

# Please specify the global repository information
DFC.GLOBAL_REGISTRIES=repo1
DFC.GR_USERNAME=dm_bof_registry
DFC.DOCBROKER_HOST=win102
DFC.DOCBROKER_PORT=1489
DFC.SECURE.GR_PASSWORD=global_repository_password

#Below 2 options are used to generate a report of profile customizations applied in
the previous installation of CTS.
#Use this report to assess the profile customizations you want to apply to the
current
```

```
installation.
#Only supported for previously installed CTS versions 7.0, 7.1 or 7.2
#The report by default is available in this location <%CTS_HOME%>/migration/log/
index.html
#Please input the product version you have previously installed. If not 7.0,7.1 or
7.2,
please leave it blank
CTS.OLD_VERSION=
#Please input the previously installed product name, ADTS, AVTS or MTS.
#If you have never installed CTS product, please leave it blank. If you have more
than
one product installed, please separate product name by comma.
CTS.OLD_PRODUCTS=
# Please specify SSL certificate information. Set "USE_CERTIFICATES" to "true" if
you
want to enable SSL non-anonymous communication.
USE_CERTIFICATES=false
DFC_SSL_TRUSTSTORE=C:\dfc.keystore
DFC_SSL_TRUSTSTORE_PASSWORD=password
DFC_SSL_USE_EXISTING_TRUSTSTORE=false

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

CTS.DOMAIN_USER_LIST=

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

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

11. Open the command prompt and navigate to the installer directory where the `rqm_config_adding.properties` file is available.
12. For Windows, run the installer silently from the command prompt, using this command:

```
rqmConfigurator.exe -f rqm_config_adding.properties
```

For Linux, run the following command:

```
rqmConfigurator.sh -f rqm_config_adding.properties
```

13. Verify the configuration logs created in the `%RQM_HOME%\logs\CTS_log` folder.

6.14.2 Unconfiguring and uninstalling Records Queue Manager services

You must first unconfigure Records Queue Manager before uninstalling it.

To manually unconfigure a Records Queue Manager instance:

1. Navigate to RQM Configurator under **Start > Programs > RQM Configuration Utility**.
2. If you want to use Vault, on the **Documentum Server RQM Vault Components** page, select the **Enable Vault** check box.
3. If you selected the **Enable Vault** check box, provide the following information and click **Next**:

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

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

- b. **DSIS Token:** Provide the `dsis.dctm.token` token value as described in [“Configuring Documentum Secret Integration Service” on page 14](#).

! Important

If you enabled the Vault configuration, the configurator retrieves all the password information automatically from Vault. This includes the `rqmSysAdminPass`, `DFC_GLOBALREGISTRY_PASSWORD`, and `INSTALL_OWNER_PASSWORD` passwords.

4. Select **Remove a Records Queue Manager instance from a repository** and click **Next**.
5. Select the repository from which you want to remove the Records Queue Manager server. Enter the SuperUser name, the SuperUser password, and the Domain if any.
6. Click **Next**.

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

7. Click **Done** in the dialog box indicating the success of the Records Queue Manager Server removal.

This instance of RQM is now removed from the repository.

To uninstall Records Queue Manager:

1. On the host machine, select **Start > Settings > Control Panel** and double-click **Add/Remove Programs**.
2. Select **Documentum Records Queue Manager**, and then click **Change/Remove**.
3. In the **Change/Remove** dialog box, click **Next**.

4. Click **Next**.
5. If you need to uninstall Foundation Java API, select **Uninstall DFC** and click **Next**. A dialog box indicates that the selected Records Queue Manager is uninstalled.

6.14.3 Unconfiguring and uninstalling Records Queue Manager services in silent mode

To unconfigure Records Queue Manager silently:

1. Navigate to %RQM_HOME%\server_install\server_install folder. This folder must contain the file `rqm_config_removing.properties`.
2. To unconfigure Records Queue Manager from a repository, update the `rqm_config_removing.properties` file with the relevant values using a text editor:

```
#all

# Please specify the config actions, "add" or "remove".
CTS.INSTALL_TYPE=remove
# Please specify the host name for CTS machine
CTS.HOSTNAME=MTS76
# Please specify the repository name
CTS.DOCBASE=repo1
# Please specify the repository super user and password
CTS.DOCBASE_SUPERUSER=Administrator
CTS.NOT_ENCRYPTED_PASSWORD=password
# Please specify the domain name (optional)
CTS.DOMAIN=
# Please specify the products which you want to remove if you install multiple
products in same machine. If you remove multiple products, use comma as delimiter.
# Config will remove all products, if this one remain empty. The value can be CTS-
Media,
CTS-Documents, CTS-Audio-Video, XTS or combination any of them.
CTS.MODULE_NAMES=

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

3. Open the command prompt and navigate to the installer directory where the `rqm_config_removing.properties` file is available.
4. For Windows, run the installer silently from the command prompt, using this command:

```
rqmConfigurator.exe -f rqm_config_removing.properties
```

For Linux, run the following command:

```
rqmConfigurator.sh -f rqm_config_removing.properties
```

To uninstall Records Queue Manager silently:

1. Navigate to the folder `C:\Documentum\uninstall\rqm`.
2. Update the `rqm_uninstall.properties` file with the relevant values using a text editor:

```
## silent uninstall response file
```

```
INSTALLER_UI=silent  
  
###uninstallation  
UNINSTALL_DFC=true
```



Note: UNINSTALL_DFC attribute is set to true by default and Foundation Java API instance will get uninstalled by setting the attribute to true.

3. For Windows, run the installer silently from the command prompt, using this command:

```
Uninstall.exe -f rqm_uninstall.properties
```

For Linux, run the following command:

```
Uninstall.bin -f rqm_uninstall.properties
```


Chapter 7

Upgrading a WDK-based application

This chapter contains instructions that are shared by all WDK-based products. Check your release notes for information on the application servers, browsers, and other software that are certified for your product. Review this chapter and perform the tasks described in it before upgrading a WDK application. Customization of Documentum Administrator is not supported.

“[Directories and files to back up](#)” on [page 95](#) shows the files, directories, and subdirectories on the application server host that should be backed up.

Table 7-1: Directories and files to back up

Directory/file	To back up if present
custom/app.xml	app.xml
custom subdirectories	JSP files
custom/config	XML files
custom/strings	Properties files
custom/theme subdirectories	Branding files
WEB-INF/classes subdirectories	Custom classes
custom/src subdirectories	Custom source files
WEB-INF/tlds	Custom tag libraries
WEB-INF/classes/com/ documentum/web/formext/session	Back up AuthenticationSchemes.properties, KeystoreCredentials.properties, and TrustedAuthenticatorCredentials.properties if customized

After upgrading, recompile your custom classes to ensure that the custom code still works. Add your backed up files to the new application for testing. The upgrade and migration instructions documented in *OpenText Documentum Content Management - On-Premises Upgrade and Migration Guide (EDCCS-UMD)* are also applicable for Retention Policy Services and Records Manager release.

Chapter 8

Deploying a WDK-based application

This chapter contains instructions that are shared by all WDK-based products. Check your release notes for information on the application servers, browsers and other software in the environment that are certified for your product.

After you complete the required predeployment tasks, deploy a WDK application on the application server host.

8.1 Preparing the WAR file for deployment

Perform the following procedure to prepare the WDK-based application WAR file.

To deploy a WDK-based application

1. Download the WDK application WAR file from My Support to a temporary directory on the application server host.
2. Unpack the WAR file and modify the `dfc.properties` file following the instructions in [“Enabling Foundation Java API connections to repositories” on page 98](#). You must perform this procedure before attempting to connect to Documentum repositories.
3. Enable the optional presets and preferences repositories in `dfc.properties` following the instructions in [“Enabling presets and preferences repositories” on page 101](#).
4. Add or migrate customizations from previous WDK-based applications.
5. Apply language packs if you have purchased them.
6. Make any UCF configuration changes that your applications need before deploying. Refer to the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* for details.
7. Re-archive the WAR file.
8. Deploy the WAR file according to the deployment instructions in your application server documentation.

8.2 Enabling Foundation Java API connections to repositories

You must provide connection broker and global registry values in `dfc.properties` before your application can connect to repositories.

A global registry of Documentum CM Server is required for WDK-based applications. The global registry is a central repository that serves several purposes:

- Deploys service-based business objects (SBOs)
- Stores network location objects
- Stores application presets, unless another repository is configured in `app.xml`
- Stores persistent user preferences, unless another repository is configured in `app.xml`

The *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)* contains information about enabling a repository as a global registry.

You can copy information from the `dfc.properties` file that was generated by the Documentum CM Server installer on your global registry host. The generated `dfc.properties` file contains the connection broker address and the encrypted global registry user login information.

To use the `dfc.properties` file information from the global registry Documentum CM Server repository:

1. On the global registry repository host, locate the Documentum CM Server installation directory. On Windows hosts, the default installation directory is `C:\Documentum`. On Linux hosts, this directory is specified by the environment variable `$DOCUMENTUM`.
2. Open the file `dfc.properties` that is located in the `config` subdirectory.
3. Copy the following keys and their values from the file:

```
dfc.docbroker.host[0]=<address>
dfc.globalregistry.repository=<repository_name>
dfc.globalregistry.username=<username>
dfc.globalregistry.password=<encrypted_password>
```
4. Unpack the application WAR file.
5. Open the `dfc.properties` file located in `WEB-INF/classes` within this expanded WAR file directory.
6. Paste in the values that you copied from the global registry `dfc.properties`.
7. Use a text editor to configure additional properties in this file or make any changes to it.

8. Save the `dfc.properties` file and deploy the application.



Note: If you create a new WAR file from this application directory, you must ensure that any paths that you specify in `dfc.properties` are valid directories on the application server and that the application server instance owner has write permission on the specified directories.

To configure connections in `dfc.properties` before deployment:

1. Unpack the application WAR file.
2. Open the file `dfc.properties` in `WEB-INF/classes`.
3. *Trusted Login* allows the user currently logged into the Documentum CM Server to connect to the repository without requiring a password. If your application server is running on Documentum CM Server as the *installation owner*, you may experience this behavior. Add the following setting in the `dfc.properties` file:

```
dfc.session.allow_trusted_login = false
```

4. Add the fully qualified hostname for the connection broker to the following key. You can add backup hosts by incrementing the index number within brackets.

```
dfc.docbroker.host[0]=<host_name>
```

5. If you wish to use a port for the connection broker other than the default of 1489, add a port key to `dfc.properties`:

```
dfc.docbroker.port=<port_number>
```

6. Add the global registry repository name to the following key:

```
dfc.globalregistry.repository=<repository_name>
```

7. Add the username of the `dm_bof_registry` user to the following key:

```
dfc.globalregistry.username=<dm_bof_registry_user_name>
```

The global registry user, who has the username of `dm_bof_registry`, has read access to objects in the `/System/Modules` and `/System/NetworkLocations` only.

8. Add an encrypted password value for the following key:

```
dfc.globalregistry.password=<encrypted_password>
```

You can either copy the username and encrypted password from the `dfc.properties` file on the global registry Documentum CM Server host, or you can select another global registry user and encrypt the password using the following command from a command prompt (assumes the directory containing `javaw.exe` is on the system path): This command needs to be run on the application server:

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

For example (Tomcat <version>), this is what the command looks like:

```
java -classpath C:\tomcat\apache-tomcat-<version>\webapps\records\
WEB-INF\lib\dfc.jar com.documentum.fc.tools.RegistryPasswordUtils dm_bof_registry
GrRNPbLJrkoTDAZEORGJow==
```

Compare with if this command is run on the Documentum CM Server (the password is much longer and cannot be used for the application)

```
C:\Documentum\java64<version>\bin\java -classpath C:\Documentum\Shared\dfc.jar
com.documentum.fc.tools.RegistryPasswordUtils dm_bof_registry
AAAAEAA0S46t3Kp+00jG96JVmdTaamBp9kG1JaosXAo/1DCh
```

9. Save the `dfc.properties` file.

8.3 Enabling Foundation Java API memory optimization

The Foundation Java API diagnostics are set to true by default. To free up memory resources, set `dfc.diagnostics.resources.enable` in `dfc.properties`. Refer to [“Enabling Foundation Java API connections to repositories” on page 98](#) for the procedure of unpacking the war file and modifying `dfc.properties`. Add the following line to your `dfc.properties` file:

```
dfc.diagnostics.resources.enable=false
```

8.4 Configuring UCF

The *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* contains the following procedures:

- How to configure different content transfer mechanisms (UCF or HTTP) for roles.
- How to configure the UCF client content transfer directories, including client path substitution.
- How to support self-signed or unsigned SSL certificates.
- How to configure the UCF server for forward and reverse proxy servers and alternative chunking.



Note: The web server associated with an application server must support chunked requests. The web server forwards HTTP requests using chunked transfer encoding, as described in the HTTP/1.1 protocol, to the back-end application server. If chunked requests are not supported then the client should use UCF alternative chunking mode.

To resolve a UCF error by disabling the ACS:

1. In the `Wdk/app.xml`, disable the `accelerated-read/accelerated-write`, so that content can be viewed using the application server, not the `acs/bocs`. Change *true* to *false* for `accelerated-read` and for `accelerated-write` as shown here:

```
<accelerated-read>
  <!-- when set to be "true",
```

```

enables ACS, and, when optimal, BOCS read operations -->
  <enabled>false</enabled>

<accelerated-write>
  <!-- when set to be "true",
enables ACS, and, when optimal, BOCS read operations -->
  <enabled>false</enabled>

```

2. Save and restart Tomcat.

8.5 Enabling presets and preferences repositories

By default, presets and persistent preferences are stored in the global repository. For better performance, you can configure your application to use different repositories for presets and persistent preferences.

Add your preferences repository settings to `app.xml` in the `/custom` directory of the application. Copy the entire `<preferencesrepository>` element from `/wdk/app.xml` into `/custom/app.xml` and then specify your repository. For information on other preferences settings in `app.xml`, refer to *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*.

Table 8-1: Preferences configuration elements

Element	Description
<code><preferencesrepository></code>	Contains a <code><repository></code> element. If this element is not present, user preferences are stored in the global repository, which can slow down performance.
<code>.<repository_path></code>	Specifies the path within the preference repository in which to store preferences. If the path does not exist at application startup, it will be created.
<code>.<repository></code>	Specifies the repository in which to store preferences, preferably not the global repository.

To give users the ability to create presets using the presets editor, assign those users the `dmc_wdk_presets_coordinator` role.

8.5.1 Configuring encrypted password for presets and preferences repositories

1. Change the default passwords in Documentum CM Server.

For both `dmc_wdk_presets_owner` and `dmc_wdk_preferences_owner` users, the default password must be changed using IAPI in Documentum CM Server. To do this, log in to IAPI as an administrator and execute the following commands.

- To change the password for the `dmc_wdk_presets_owner` user:

```
retrieve,c,dm_user where user_name='dmc_wdk_presets_owner';
set,c,1,user_password
<enter new password>
save,c,1
```

- To change the password for the `dmc_wdk_preferences_owner` user:

```
retrieve,c,dm_user where user_name='dmc_wdk_preferences_owner';
set,c,1,user_password
<enter new password>
save,c,1
```

2. Encrypt the passwords in Records Client using the `TrustedAuthenticatorTool` available in the `WEB-INF/classes` folder.

Windows:

To create an encrypted password, execute the following command at the command prompt:

```
java -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.base/java.lang=ALL-UNNAMED -cp ./lib/*; TrustedAuthenticatorTool <PASSWORD>
```

where *<packaged-version>* is the supported version packaged with the Records Client WAR file.

The utility sends the encrypted password to the standard output. For example:

```
C:\records\WEB-INF\classes>java -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.base/java.lang=ALL-UNNAMED -cp .; TrustedAuthenticatorTool webtopUser@12345 Encrypted: [cQHQ//55N380I5M/Evy1ssVj0gptM+9F0IXGM/Iqpb0=], Decrypted: [webtopUser@12345]
```

Linux:

1. Navigate to the `WEB-INF/classes` folder.
2. Set the classpath for the referenced jars:

```
export JAR_PATH=../lib/dfc.jar:../lib/commons-io-<packaged-version>.jar:../lib/commons-lang-<packaged-version>.jar
```

3. Execute the Java command to generate the encrypted password:

```
java -cp $JAR_PATH TrustedAuthenticatorTool trusted
```

3. Update the encrypted passwords in `app.xml`.

Search for `<presets>` and update the `<password>` attribute with the encrypted password. For example:

```
<presets>
...
<password>cQHQ//55N380I5M/Evy1ssVj0gptM+9F0IXGM/Ipqb0=</password>
...
</presets>
```

Search for `<preferencesrepository>` and update the `<password>` attribute with the encrypted password. For example:

```
<preferencesrepository>
...
<password>cQHQ//55N380I5M/Evy1ssVj0gptM+9F0IXGM/Ipqb0=</password>
...
</preferencesrepository>
```

4. Delete the Documentum folder in `<WebApp Root>\WEB-INF\classes` (Windows) and `<WebApp Root>/WEB-INF/classes` (Linux).
5. Start the application server.

8.6 Enabling retention of folder structure and objects on export



Caution

If your users will be exporting folders with special characters (`\ / : * ? " < > |`) in its name, you should not turn this feature on. When a folder with a name containing a special character is encountered during export, an error occurs and the export fails.

To enable retaining the same folder structure (as the one in the repository) and the contained objects on the local file system when the parent folder is exported, add the following element to your `app.xml` in the custom directory:

```
<deepexport>
  <enabled>true</enabled>
</deepexport>
```

The default is false.

8.7 Working with email messages in WDK-based applications

From Records Client 20.4 onwards, there has been significant changes in the email processing functionality. The *OpenText Documentum Content Management - Records Client User Guide (EDCRM-UGD)* provides more information. For more information on how to work with new emails and old emails, refer to [“Working with the new email processing functionality” on page 53](#).

8.8 Enabling external searches

To allow users to search external sources, an administrator must configure a connection to an ECI Services server. The ECI Services server is a separate product that is purchased separately from Webtop and Documentum CM Server. If this connection has not been configured, you cannot include external sources in your search.

8.8.1 Configuring the connection to the search server

The following procedure describes how to enable the ECI Services server to query external sources. The ECI Services documentation provides more information about how to configure the ECI Services server itself.

To configure the connection to an ECI Services server:

1. Unpack the client application WAR file.
2. Open the file `dfc.properties` in `WEB-INF/classes`.
3. Enable the ECI Services server by setting the following:
`dfc.search.ecis.enable=true`
4. Specify the RMI Registry host for the ECI Services server by setting the following:

`dfc.search.ecis.host=<host_IP>`

`dfc.search.ecis.port=<port>`

Where

- `<host_IP>` is IP address or machine name of the ECI Server.
- `<port>` is the port number that accesses the ECI server. The default port is 3005.

8.8.2 Configuring the connection to the backup search server

You can set a backup server in case the primary ECI Services server is unreachable. If a Foundation Java API-application cannot connect to the primary ECI Services server to query external sources, the backup server is contacted. You can define the time period after which the application will try to connect again to the primary server. To define the backup server, specify the RMI host and port in the `dfc.properties` file:

- `dfc.search.ecis.backup.host` : host of the backup ECI Services server. Default value is: `localhost`.
- `dfc.search.ecis.backup.port` : port of the backup ECI Services server. Default value is: `3005`.
- `dfc.search.ecis.retry.period` : waiting period before retrying to connect to the primary ECI Services server. This time is in milliseconds. Default value is: `300000`.

Chapter 9

Completing the deployment

After you deploy a WDK application, there are additional procedures that you may need to perform in order to finish and verify the deployment. Check your release notes for information on the application servers, browsers and other software in the environment that are certified for your product. This chapter contains instructions that are shared by all WDK-based products.

9.1 Deploying default virtual link support

A virtual link is a URL that resolves to a document in a repository. The virtual link URL contains the repository name, folder path, and object name of the content to be accessed. All WDK-based applications support virtual links in the following form:

```
http(s)://server:port/app-name/repository-name:/folder-path/.../objectname
```

You can install default virtual link support for URLs that do not contain the web application names. These links will be redirected to the current application. Default virtual links URLs have the following form:

```
http(s)://server:port/repository-name:/folder-path/.../objectname  
http(s)://server:port/RightSite/repository-name:/folder-path/.../objectname  
http(s)://server:port/rs-bin/RightSite.dll/folder-path/.../objectname
```

To install default virtual link support:

1. Deploy the `vlink.war` file as the root web application on the application server.
Some application servers have an existing root web application which you must replace with the default virtual link application. Others require you to create a root web application manually or during application server installation. Refer to the documentation for the application server for information on a root web application.
2. Deploy the virtual link war file (`vlink.war` or `ROOT.war`) to the application server by using the mechanism recommended by the application server for deploying a default web application.
3. Modify the **DefaultWdkAppName** param-value in the `web.xml` of the virtual link WAR file. This parameter value specifies the WDK-based application that will handle the virtual link request if there is no current repository session for the user. If you do not specify a parameter value, it will default to **webtop**.

For more information on virtual links, refer to the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*.

9.2 Accessing the application

This section provides you with information on accessing and testing the deployment of a WDK-based application by connecting through a browser client. Before you test the deployment, ensure the application is started in the application server. For information on starting the application, refer to the documentation of the application server.

If the application requires additional configuration or setup, such as installing a DAR or DocApp, perform those steps before you test the application.

To verify the deployment and configuration of a WDK application:

1. Open a browser window and type this URL:

```
http://<host_name>:<port_number>/<virtual_directory>
```

Where:

- *<host_name>* is the host where the application server is installed. If the browser is on the application server machine, substitute localhost for *<host_name>*; for example, `http://localhost:8080/records`.
- *<port_number>* is the port where the application server listens for connections
- *<virtual_directory>* is the virtual directory for your application

For example, if the application server host is named iris, the port is 8080, and the application virtual directory is webtop, the URL is `http://iris:8080/records`.

2. Log in to a repository through the WDK-based application.

If the login succeeds, the application is correctly deployed and configured.

9.3 Testing WDK samples

After deploying a WDK-based application, you can view WDK sample pages after logging into a repository. The sample JSP pages, component definitions, and supporting compiled class files are provided in a zip file along with the product download. Unzip them to your application root directory, preserving the folder hierarchy in the zip file.

To view the WDK samples:

1. Ensure that the application server is running.
2. Open a browser and type the following URL:

```
http://<host_name>:<port_number>/<virtual_directory>/component/login
```

Where:

- *<host_name>* is the host where the application server is installed

- `<port_number>` is the port where the application server listens for connections
- `<virtual_directory>` is the virtual directory for the application

A login dialog box appears.

3. Log in to a test repository.

The login dialog box reappears with the status message **Login Successful**.

4. Type this URL:

```
http://<host_name>:<port_number>/<virtual_dir>/wdk/samples/index.jsp
```

This page displays a list of the available samples.

5. Click **Session Zoo** and type a valid repository username, password, repository name, and domain (if required), then click **Create Connection**.

The repository is listed in the **All Connected Repositories** section of the page, and the Status message line starts with Successfully connected to repository `<repository_name>`

6. Continue to experiment with other samples, especially Menu Zoo, Tree Control, and FX Control Pens.

Some samples have **Create Test Cab** and **Destroy Test Cab** buttons. These create and delete a test cabinet in the repository and require Create Cabinet privileges.

9.4 Configuring user authentication

Refer to the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* for the following:

- Single Sign-On for Security servers
- OpenText Directory Services (OTDS)

For configuring OTDS, see *“Configuring OpenText Directory Services”* on page 16.

9.5 Licensing OpenText Documentum CM

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

Chapter 10

Installing Application Connectors

10.1 Overview

Application Connectors provide users with the ability to access a repository directly from content authoring applications. For example, a user writing a document with Microsoft Word can check the document into the repository from within Word. The modal dialog window does not display the frameset of Webtop or other WDK client application.

The Application Connectors installer runs on the client in one of two ways:

- GUI installation
The administrator notifies the end user to install Application Connectors. The email contains the URL to the installer. The installer is part of the WDK application, in the path `/webcomponent/install/appconnectors`.
- Command-line installation
Microsoft Systems Management Server (SMS) is used to distribute Application Connectors to Microsoft Office users with an Microsoft Installer (MSI) based installer.

Application Connectors work with UCF content transfer only.

The Application Connectors installer disables Documentum Desktop Office integrations before installing Application Connectors. The Desktop Office integrations are disabled by removing relevant add-in files and registry entries.

The installer executable is the `Documentum-AppConnectors-Client.exe` file. When the installer is run, it verifies on each client host that the following requirements are met:

- The correct versions of the operating system and Office applications are present on the host.
- The user who installs Application Connectors is a power user or administrator.
- Sufficient free disk space is available for the installation.

The client software footprint is approximately 5 MB, with an additional 25 MB for installing the .NET framework if it is not already present on the client.

Supporting Windows software, such as .NET, is installed if it is not already installed on the client host.

The Application Connectors installer will upgrade an existing installation on the client. No uninstall of the previous version is necessary. Application Connectors

version 6.x is required in order to connect to Webtop version 6.x. It can also connect to Webtop applications of version 5.3.x.

10.2 GUI installation of Application Connectors

Ensure that the Webtop-based is running and available when you run the Application Connectors installer so that the menu for the authoring application can be downloaded from the Webtop-based application.

There are two methods of launching the GUI installer on the client. You can use the general application installer utility or download the Application Connectors installer and run the installer from the local disk.

To download and install Application Connectors on the client host:

1. Log in to the client host as a user with power user or administrator privileges.
2. Uninstall previous installations of Application Connectors.
3. Close any running Microsoft Office applications, whether they are running as standalone applications or as instances within Outlook.
4. Open a browser session and type the URL to the installer. The URL is typically provided by an administrator.

A dialog box appears, asking whether to save the file or run the file.

To create the URL for users to install Application Connectors, replace webtop with the application alias:

```
http://<hostname>:<port>/webtop/webcomponent/install/  
appconnector/Documentum-AppConnectors-Client.exe
```

5. Click **Install**.
6. Click **Save** and download the file to your desktop.
7. Double-click the saved file to begin installation.
A welcome screen appears with a warning that the installer disables Desktop Client if it is found.
8. Click **Next**.
The Customer Information Dialog appears.
9. Choose **Current User** or **All Users**.
10. Click **Next**.
11. For **Enter URL**, type the URL to the Webtop-based installation that you will use within the Office applications, for example:

```
http://plegion:8080/webtop
```

12. Click **Next**.

13. Click **Install** to launch the installer.

After installation has completed, the Documentum menu is available within the authoring application for which an Application Connector was installed. Figure 10-1 shows the Documentum menu within Microsoft Word.

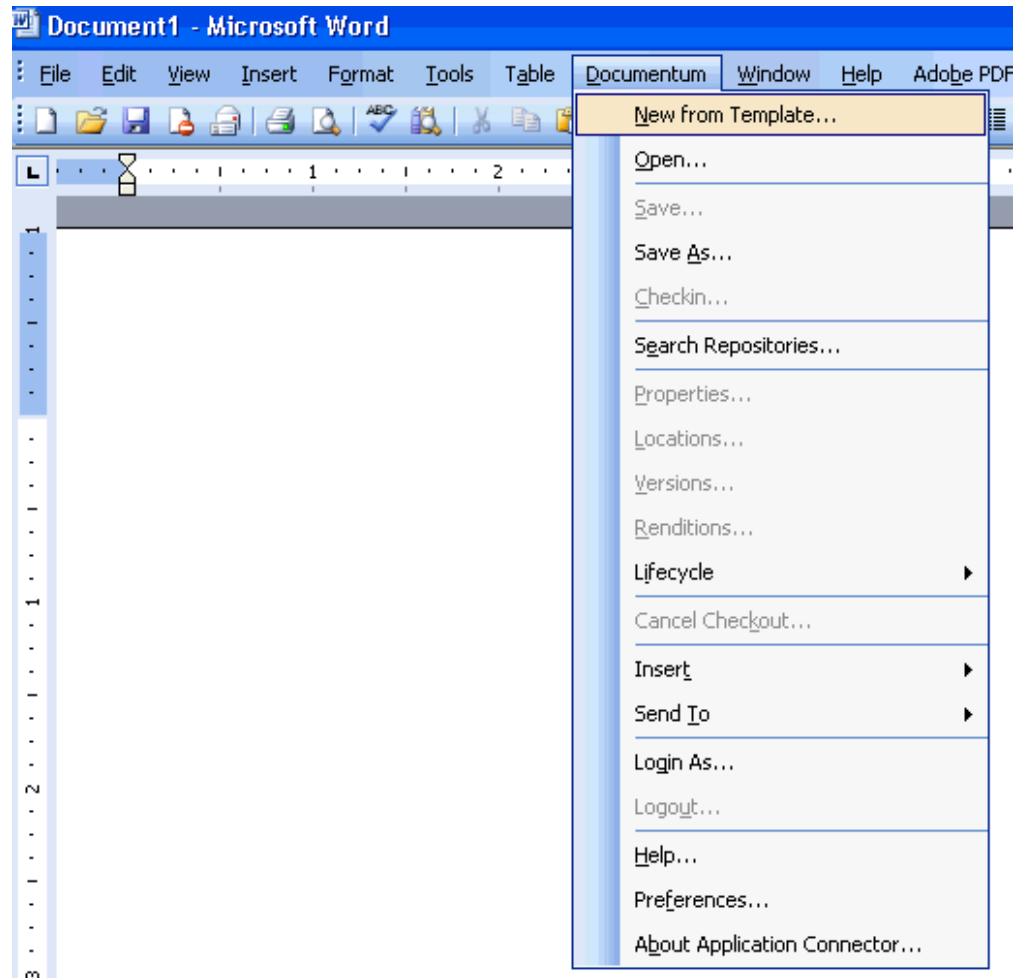


Figure 10-1: Documentum menu in authoring application



Note: Only one Webtop-based application can be used by Application Connectors at a time. To change the URL to a different Webtop application, open the **Documentum** menu in the authoring application and choose **Preferences**. Copy the new URL into the **URL** text box.

10.3 Commandline installation of Application Connectors

The MSI installers are located within the Webtop-based application in the folder /webcomponent/install/appconnector.

The following examples illustrate the use of standard command-line parameters for a Windows installer. Information about these parameters can be found in the Microsoft MSDN Library. Line breaks have been introduced into the example for readability only. Do not use line breaks when you issue these commands from the command line. Substitute your application server alias and port, if needed, for **server** in the examples.

Running the installer from the command line

Here is the syntax to run the installer in command-line mode:

```
Documentum-AppConnectors-Client.exe /v"WEBTOPURL=http://server/folder"
```

Running the installer in silent mode

The following syntax launches the installer silently from the command line:

```
Documentum-AppConnectors-Client.exe /s /v"/qn WEBTOPURL=http://server/appname"
```

Changing the Documentum menu name during installation

The following syntax changes the menu name to "MyCompany". The menu name should have no spaces, and you must enter the command without a line break:

```
1 Documentum-AppConnectors-Client.exe /s /v"  
2 /qn WEBTOPURL=http://server/appname MENU_NAME=MyCompany"
```

Deleting Normal.dot during installation

A command-line option forces the installer to delete the Normal.dot file created by Microsoft Office. You may want to do this if you are installing to machines that previously had Documentum Desktop installed and did not have customizations in Normal.dot. To delete Normal.dot in silent mode, enter the following command without a line break:

```
Documentum-AppConnectors-Client.exe /s /v"/qn  
WEBTOPURL=http://server/webtop DELETE_NORMAL_DOT_DOT=TRUE"
```

Chapter 11

Installing Documentum Webtop Extended Search

Documentum Webtop Extended Search provides the following additional search capabilities for any WDK-based application:

- *Clustering*—Results are automatically and dynamically grouped into categories.
- *Search templates*—Users can reuse queries saved with predefined constraints and only need to set the variable fields.
- *Search monitoring in real time*—Status data related to the returned results are updated in real time for each source.

The extended search features are available, which requires the installation of DARs in addition to the WDK-based application. The ECI Services server is not required to support Webtop Extended Search. It is only required to search external sources and is installed separately.



Note: No license file is required for the DAR installation.

To enable clustering, use the Webtop Extended Search installer to deploy the Clustering DAR to a global registry repository. This will also enable search monitoring on every web application that uses the global repository. Deploy the Search Templates DAR on each repository in which you want users to save Search templates.

When you run the installer, it presents a checklist of repositories that are available to the connection broker. The Clustering DAR will be deployed to any global registry repositories that you select, and the Search Templates DAR will be deployed to all repositories that you select.

You can launch the installation on the machine hosting the Documentum CM Server or on another machine from which the Documentum CM Server is visible. The operating system of the machine from which you launch the installation must be Windows or Linux. The following procedure describes the installation steps.



Note: You must have superuser privileges for the repositories in which you want to install the DARs.

11.1 Installing the Webtop Extended Search option

1. Download the installer file corresponding to your operating system from My Support. You may also download the ZIP file that contains the language pack(s) for languages other than English. These language pack(s) are necessary to display localized Search templates.
2. Unzip the downloaded file to a temporary directory. The directory should contain:
 - common files for all operating systems:
 - brand.jar
 - composer.jar
 - dars.jar
 - darSetup.jar
 - specific files according to the operating system:
 - dar<Operating_system>SuiteSetup.jar
 - dfc<Operating_system>Setup.jar
 - The installer file: dar<Operating_system>SuiteSetup.exe or dar<Operating_system>SuiteSetup.bin

where <Operating_system> is the abbreviation or the name of the operating system, such as Win for Windows, or Linux.
3. For Linux operating systems, create the following environment variables, which are required by the installer. If they already exist on the system host, you can skip this step:
 - DOCUMENTUM
Specifies the full path of the user root directory. Can be any directory in the installation owner's environment, for example:

```
DOCUMENTUM=/export/home/Documentum
```
 - DOCUMENTUM_SHARED
Specifies the Foundation Java API program root directory. Can be any directory in the installation owner's environment, for example:

```
DOCUMENTUM_SHARED=$DOCUMENTUM
```
4. Double-click on the installer file that corresponds to your operating system to launch the installation.

The **Welcome** screen lists the products that will be installed:

 - DAR Deployer <version>

- DFC Runtime Environment <version>
5. Click **Next** to continue.
The **License agreement** screen appears.
 6. Click **I accept the terms of the license agreement**, and then click **Next**.
The **Select Optional Features** screen offers the possibility to install optional features for Foundation Java API, in this installation, you can install the Foundation Java API Developer Documentation (javadocs).
 7. Click **Next** to continue.
 8. In the **Connection Broker** screen, enter a **Connection Broker Host Name** and **Connection Broker Host Port**. The connection broker should have access to the repositories on which you want to install the DARs. If the repositories are not visible, you have to run another installation specifying another connection broker. Click **Next** to continue.
The **Select repositories** screen appears.
 9. Select the repositories in which the DARs will be deployed. The Clustering DAR will only be deployed in repositories that are configured as global registries. The Search Templates DAR will be deployed in all the repositories that you select.
After you select repositories, click **Next** to continue.
The **Repository Login** screen appears.
 10. Enter the **User name**, **Password** and optional **Domain** for each repository you previously selected.
Click **Next**, and the credentials are tested against the repositories.
The following screen indicates the directory path where the DARs will be deployed. No files are installed in the local system but an install log is created.
 11. Click **Next** to continue.
 12. The DARs are installed. When the installation is complete, click **Finish**.



Note: To get localized Search templates, you need to install the corresponding Language Pack(s) using Documentum Composer. First download the Language Packs ZIP file from My Support then refer to Documentum Composer documentation for more information about the installation procedure of a DAR file with localized resource files.

11.2 Viewing installation log files

Log files are automatically created for the installation of the DARs. To view the installation log files, navigate to the installation directory, and locate the subdirectory `dm_log`. This directory contains `SearchTemplates_installerlog.html`, and it contains `Clustering_installerlog.html` if the clustering feature has been installed.

Chapter 12

Enabling the Webtop Express DAR

Documentum CM Server installs the Webtop Express DAR. This DAR creates lightweight functionality for an Express user by means of presets. To make this functionality available, add users to the Express User (express_user) role. This role is installed by the Webtop Express DAR.

“Express user capabilities” on page 119 describes the functionality that is available to Webtop Express users.

Table 12-1: Express user capabilities

Preset	Values
Formats	None Text PDF all MS Office formats
Types	dm_document dm_folder
Templates	Displays templates that correspond to formats
Actions	Document: Content transfer, subscriptions, email, quickflow, Properties, clipboard actions, create, delete Excluded: Relationships, export to CSV, favorites, notifications, lifecycle and virtual document actions, tools (most); new workflow template, room, form, cabinet
Locations	Home Cabinet Cabinets Subscriptions Recent Files Inbox (not Searches, Categories, Administration)

Preset administrators who belong to the dmc_wdk_presets_coordinator role can change the enabled or excluded features and allowable values by editing the Webtop Express preset in the Presets Editor UI.

Chapter 13

Troubleshooting deployment

This chapter contains information on troubleshooting a WDK application deployment. Not all items may apply to your WDK-based product or environment. Refer to the deployment guide and the release notes for your specific WDK application for information regarding additional items that can affect deployment, configuration and usability.

13.1 Wrong JRE used for application server

If the application server host has multiple JREs on the system, the wrong JRE may be used by the application server. Check your application server documentation for instructions on using the correct JRE with your application server. For example, the Tomcat application server uses a JAVA_HOME environment variable. If this variable value is specified in the application startup batch file `catalina.bat` or in the `service.bat` file for Windows services.

The error that is displayed in Tomcat using the wrong JRE is the following:

```
ERROR [Thread-1]
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/webtop]
- Error configuring application listener of class
com.documentum.web.env.NotificationManager
java.lang.UnsupportedClassVersionError:
com/documentum/web/env/NotificationManager
(Unsupported major.minor version 49.0)at
java.lang.ClassLoader.defineClass0(Native Method)
```

13.2 No global registry or connection broker

Global registry information must be configured in `dfc.properties`. The application server must be able to download required BOF modules from the global registry repository. If the information in `dfc.properties` is incorrect, the application server cannot download appropriate BOF modules, and following exception is thrown:

```
ERROR...Caused by: DfDocbrokerException:: THREAD: main; MSG:
[DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "10.8.3.21:1489" failed;
ERRORCODE: ff; NEXT: null
```

To fix this error, either provide the correct BOF registry connection information in `dfc.properties`, or do not provide any connection information at all. Refer to the *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY-IGD)* for information on enabling a repository as a global registry.

13.3 No connection to repository

If the application server log contains the following error during application initialization, it indicates that you have not specified a connection broker in the `dfc.properties` file of your application WAR file:

```
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:432)
Caused by: DfDocbrokerException:: THREAD: main; MSG:
[DFC_DOCBROKER_REQUEST_FAIL
ED] Request to Docbroker "10.8.3.21:1489" failed; ERRORCODE: ff;
NEXT: null
```

A WDK-based application must have information about the available connection broker in order to establish a connection to repositories. Refer to [“To configure connections in `dfc.properties` before deployment:” on page 99](#) for information on enabling the connection in `dfc.properties`.

If the repository that is specified as the global repository is down, the following message may be displayed:

```
Caused by: DfNoServersException:: THREAD: main; MSG:
[DM_DOCBROKER_E_NO_SERVERS_FOR_DOCBASE]error: "The DocBroker running
on host (10.8.3.21:1489) does not know of a server for the specified
docbase
(wtD6winsql)"; ERRORCODE: 100; NEXT: null
```

13.4 Login page incorrectly displayed

If the login page displays several login buttons, the browser does not have the Java plug-in installed. You must download and install the Java plug-in for the browser.

If the login page displays several controls with the same label, you have not turned off tag pooling in the application server. Refer to [“Tag pooling problem” on page 123](#) for troubleshooting information on this problem.

13.5 Slow performance

Many performance enhancements are documented in *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)*. Set `dfc.diagnostics.resources.enable` to `false` in `dfc.properties` unless you are using the Foundation Java API diagnostics. This setting uses a significant amount of memory.

13.6 Out of memory errors in console or log

Check to make sure that you have allocated sufficient RAM for the application server VM. For more information, refer to [“Setting the Java memory allocation” on page 23](#).

The following error is common when the MaxPermSize is set too low:

```
java.lang.OutOfMemoryError: PermGen space
```

13.7 Slow display first time

The first time a JSP page is accessed, it must be compiled by the application server. It is much faster on subsequent accesses.

If you have tracing turned on, or if you have a very large log file (of several megabytes), the browser response time dramatically decreases.

13.8 Foundation Java API using the wrong directories on the application server

If you have not specified content transfer directories in dfc.properties, Foundation Java API will first look for global environment variables that set directory locations.

13.9 Tag pooling problem

If you have not properly disabled tag pooling in the application server, you will see several instances of the same control on the login page. For instructions on disabling pooling in Tomcat, refer to [“Preparing Tomcat” on page 32](#).



Caution

After you disable tag pooling, you must clear the cached JSP class files which still may contain pooled tags. Refer to your application server documentation to find the location of the generated class files. For example, Tomcat displays the following error message:

```
com.documentum.web.form.control.TagPoolingEnabledException: JSP tag pooling is not supported.
```

13.10 UCF client problems

Refer to the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* for more details.

13.11 Citrix client problems

On the Citrix Server, ensure that the WDK-based application is published, the Citrix desktop is published, and the user's roaming profile is set up correctly so that UCF will not download to the local host. Perform the following procedure to clean up UCF for roaming users if the roaming profile was not set up properly.

To configure the web application for roaming profiles

1. Delete the OpenText Documentum CM directory that was installed in the user's home directory, for example, C:\Documents and Settings\<user name>\Documentum.
2. Edit `ucf.installer.config.xml` in `/wdk/contentXfer` in the WDK application. Change every environment variable in this file that uses the Java home directory `$java{user.home}` to use the roaming profile environment variable:

```
<defaults>
  <ucfHome value="$env(USERPROFILE)/Documentum/ucf" />
  <ucfInstallsHome="$env(USERPROFILE)/Documentum/ucf" />
  <configuration name="com.documentum.ucf">
    <option name="user.dir">
      <value>$env{USERPROFILE}/Documentum</value></option>
```

3. Save and restart the application server.

13.12 Connection issues between an ECI Services server and IPv6 clients

ECI Services server uses the RMI protocol to communicate with the client applications. When the client application launches a request against the ECI Services server, it indicates the IP address that the ECI Services server should use to respond. However if the client has multiple IPs, it may send an IP address that the ECI Services server cannot use to respond. To avoid any connection issue, you need to modify the command that launches the client by setting the `Djava.rmi.server.hostname` property in the Java options.

The following example describes how to update the `catalina.bat` script that launches the WDK application and forces the RMI IP to connect:

```
set JAVA_OPTS=%JAVA_OPTS% -Djava.rmi.server.hostname=<IPv6 address>
```

13.13 Presets not working

Presets may not work if you start the application server before starting the repository in which your presets are stored because the WDK application might have requested the presets from the repository, which had not been initialized completely. Check the application server logs for a connection failure while loading presets. To resolve this, restart the application server.

Chapter 14

Deploying a custom application

The following topics describe tools that assist in packaging your custom application. Refer to your product release notes to determine whether customization of the product is supported.

14.1 Using the comment stripper utility

Your JSP pages will load faster if you strip out white space and comments. A comment stripper tool, CommentStripper, is provided in `/WEB-INF/classes/com/documentum/web/tools`. [“Comment stripper utility parameters” on page 127](#) describes the parameters to use in starting this tool from the console.

Table 14-1: Comment stripper utility parameters

Parameter	Description
<code>args filename</code>	Removes comments from a single file
<code>args *.ext</code>	Removes comments from all files with the specified extension
<code>?</code>	Displays help
<code>l</code>	Removes leading white space
<code>t</code>	Removes trailing white space
<code>m</code>	Removes HTML comment blocks <code><!--...--></code> and <code><!--...--></code>
<code>j</code>	Removes JSP and JavaScript <code>/ * ... */</code> comments
<code>r</code>	Recurses directories from current
<code>oxx</code>	Uses specified extension instead of overwriting original file
<code>v</code>	Outputs in verbose mode (OFF by default)

The tool has already been run on some WDK-based applications such as Webtop. The commented files, useful for development, are provided in a JAR file in the base directory: `unstripped.jar`.

Chapter 15

Installing Brava

The Brava! Enterprise HTML viewer (which is integrated with Brava! Enterprise Server), provide users the ability to load documents from the Brava! Server online using a web browser. Documents are processed with the Brava! Enterprise Server as HTML output and are presented in this streamlined viewer for quick viewing, comparing, searching, publishing, redacting, and adding/reviewing annotations.

To install Brava! Enterprise for the Records Client:

1. Follow the *OpenText Documentum Web Development Kit and Webtop - Deployment Guide (EDCCLWT-AGD)* instructions and make sure to type records (or the URL you prefer to use) instead of Webtop when instructed.
2. Modify XML files as instructed according to the following substeps:
 - a. Extract recordsbrava_<version>.zip into the webapps\records folder.
 - b. Update the custom\app.xml file to extend **bravaweb** instead of **rm**.
 - c. Update the bravaweb\app.xml file to extend **rm** instead of **webtop**.
 - d. Navigate to \webapps\records\bravaweb\config\brava.
 - e. Edit the brava_singleSelect_dm_sysobject_configuration.xml file, installed by Brava, to extend **rm**. Do not read anything into the Physical Records Manager entry. Brava is usable even if the DAR is not installed.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<config version="1.0">
  <scope>
    <menuconfig modifies="context-menu:rm/config/actions/
dm_sysobject_actions.xml">
      <insertafter path="actionmenuitem[action=export]">
        <actionmenuitem dynamic="singleselect" name="view_brava"
nlsid="MSG_BRAVA" action="brava" showifinvalid="true" />
        <actionmenuitem dynamic="singleselect" name="brava_compare"
nlsid="MSG_BRAVA_COMPARE" action="brava_compare" showifinvalid="false" />
        <actionmenuitem dynamic="singleselect" name="brava_compare_to"
nlsid="MSG_BRAVA_COMPARE_TO" action="brava_compare_to" showifinvalid="true" />
        <!-- remove this line to enable Brava! flash viewer <actionmenuitem
dynamic=
'singleselect' name='brava_flash' nlsid= 'MSG_BRAVA_FLASH'
action='brava_flash'
showifinvalid='true' /> remove this line to enable Brava! flash viewer -->
      </insertafter>
    </menuconfig>
  </scope>
</config>
```

- f. Edit the brava_singleSelect_menu_configuration.xml file, also installed by Brava, to extend **rm**. Do not read anything into the Records Manager entry. Brava is usable even if the Records Manager DAR is not installed.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<config version="1.0">
  <scope>
    <menuconfig modifies="menubar_file_menu:rm/config/library/menubar/
```

```
menubar_component.xml">
  <insertafter
    path="menu[name=file_menu].actionmenuitem[name=file_saveas]">
    <actionmenuitem dynamic="singleselect" name="view_brava"
      value="Annotate" action="brava" showifinvalid="true" />
    <actionmenuitem dynamic="singleselect" name="brava_compare"
      value="Compare To Version" action="brava_compare" showifinvalid="true" />
    <actionmenuitem dynamic="singleselect" name="brava_compare_to"
      value="Compare To..." action="brava_compare_to" showifinvalid="true" />
    <!-- remove this line to enable Brava! flash viewer
  <actionmenuitem dynamic="singleselect" name="view_brava_flash"
    value="Brava! Flash" action="brava_flash" showifinvalid="true" />
  remove this line to enable Brava! flash viewer -->
    </insertafter>
  </menuconfig>
</scope>
</config>
```

3. If Brava! server and Records Client or Webtop Client server are on different systems, then replace default IP to IP for Records Client application server.
 - a. Stop the application server.
 - b. Modify the parameters in the `brava_parameters.properties` file:

```
# IP of machine requesting a sessionId. Default set to localhost.
legal.ip.to.get.sessionid=10.5.*.* //Put a comment (#) to this line
# IP of machine requesting a sessionId for Records Client or Webtop Client
legal.ip.to.get.sessionid= //Add IP address for Records Client or Webtop Client
For example:
# IP of machine requesting a sessionId. Default set to localhost.
#legal.ip.to.get.sessionid=10.5.*.*
# IP of machine requesting a sessionId for Records Client or Webtop Client
legal.ip.to.get.sessionid=10.5.145.137
```

- c. Start the application server.

Appendix A. Predeployment checklist

Use this checklist to ensure you have performed all required tasks when you deploy or upgrade a WDK-based application.

Table A-1: Predeployment tasks

Requirement	For More Information	Completed?
Review the release notes for the release you are installing or to which you are upgrading.	The release notes are available on My Support.	
Validate your hardware configuration.	Release Notes	
Validate your application server and clients operating systems.	Release Notes	
Create any required operating system accounts.	Network administrators	
Verify that the application server instance owner has write permissions on the temporary content transfer directories.	Network administrators. The requirement is described in <i>“Content transfer directory permissions”</i> on page 13.	
Determine the repositories to which end users of the application will connect.	Network administrators	
Determine the connection brokers to which the repositories project.	Network administrators	
Determine which repository on the network is the global registry repository, and obtain the global registry user's user name and password.	Network administrators	
Determine which repositories will be used to store presets and user preferences.	Network administrators	
Determine whether language packs will be required.	Product <i>Release Notes</i>	
Prepare the application server host and application server software according to the vendor's requirements.	Specific requirements are described in <i>“Preparing the application server host”</i> on page 23.	

