

opentext™

OpenText™ Documentum™ D2

Version 16.4

Installation Guide

Legal Notice

This documentation has been created for software version 16.4.

It is also valid for subsequent software versions as long as no new document version is shipped with the product or is published at <https://knowledge.opentext.com>.

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>

Copyright © 2018 Open Text. All Rights Reserved.

Trademarks owned by Open Text.

Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries.

One or more patents may cover this product. For more information, please visit, <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

Preface	7
Chapter 1 Getting Started	9
Know Before You Start	9
Preparing for Installation	9
Instructions for Installing D2	10
Understanding the D2 Keystore Utility	12
Running the D2 Keystore Utility	13
Registering D2 DFC Clients as Privileged Clients.....	15
Chapter 2 Upgrading D2	17
Planning Your D2 Upgrades	17
Running the D2 Migration Utility	18
Upgrading D2 4.6 to D2 16.4 on the Documentum Server	21
Re-running the Upgrade Utility	24
Upgrading D2 on the Web Application Server	25
Removing Lockbox Artifacts	26
Chapter 3 Installing D2 on the Documentum Server	29
Installing D2 on the Documentum Server	29
Chapter 4 Deploying D2 DAR Files Manually	33
Deploying D2 DAR Files Manually	33
Chapter 5 Installing D2 on the Web Application Server	35
Limitations of D2 Core Installer	35
Installing D2 on Apache Tomcat for Microsoft Windows	35
Installing D2 on Apache Tomcat for a Linux Environment.....	36
Installing D2 on IBM WebSphere	37
Installing D2 on Oracle WebLogic.....	39
Installing D2 on Redhat JBOSS	40
Installing D2 on VMware vFabric tcServer for Microsoft Windows	41
Installing D2 REST Extension Framework	42
Configuring D2 REST Extension Framework	43
Chapter 6 Deploying D2 in a Docker Environment	45
Preparing the D2 Docker Environment.....	45
Running the D2 Docker Containers.....	45
Add Webapp as DFC Privileged Client	45

Chapter 7	Configuring D2	47
	Installation Validation.....	47
	Configuring D2 Config.....	54
	Configuring D2 Client.....	57
	Configuring D2 Java Method Server.....	68
	File Transfer Modes	68
	Installing DCM Browser Extension (Internet Explorer).....	69
	Installing DCM Browser Extension (Firefox).....	69
	Installing DCM Browser Extension (Chrome)	70
	Installing DCM Browser Extension (Safari).....	70
	Uninstalling Documentum Client Manager Browser Extensions	71
	Uninstalling Documentum Client Manager Native Application	71
	Configuring File Transfer Modes	72
	Configuring logback.xml for the Documentum Server.....	74
	Removing Debug Logs from the JBoss 7.1.1 JMS Log.....	75
	Configuring Documentum Server server.ini	75
	Configuring D2 Auditing	76
	Configuring Application Server Pooling Session	76
	Configuring Compatibility with Documentum Information Rights Management (IRM).....	76
	Configuring the D2EventSenderMailMethod.....	77
	Configuring D2 to use CTS for Fast Web-Enabled C2 Renditions.....	77
	Enabling POP3S Mail Configuration	78
Chapter 8	Best Practices	81
	Enabling Compression at the Application Server when using Apache Tomcat or JBoss	81
	Optimizing Performance for Widgets and Large Numbers of Content	81
	Improving Content Transfer Performance	82
	General Tuning Tips	83
Chapter 9	Configuring Authentication	85
	Single Sign On (SSO) Authentication for D2	85
	Configuring Microsoft Windows NT Unified Logon (NTLM)	85
	Configuring Kerberos	86
	Configuring Kerberos for Multiple Domains.....	88
	Configuring TrustedReverseProxy for Various SSO Environments	89
	Adding Administrator Credentials to the D2 keystore for SSO.....	91
	Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server	92
	Configuring the shiro.ini file for Custom SSO Integrations	92
	Configuring Security Assertion Markup Language (SAML) Support.....	93
Chapter 10	Installing Advanced Publishing Services	97
	Advanced Publishing Services Plug-in	97

	Deploying Advanced Publishing Services Automatically	97
	Installing the Advanced Publishing Services Plug-in	98
	Installing Advanced Publishing Services Libraries on a Linux Environment	99
	Installing Advanced Publishing Services Libraries on Microsoft Windows	100
	Installing D2-Link	101
	Troubleshooting the Installation	102
Chapter 11	Installing Advanced Publishing Services Manager	103
	Advanced Publishing Services Manager	103
	Pre-requisites for Advanced Publishing Services Manager	103
	Installing Advanced Publishing Services Manager	104
	Configuring Advanced Publishing Services Manager to run in Console Mode	105
	Viewing log information	105
	Running Multiple Instances of Advanced Publishing Services Manager	107
	Launching Advanced Publishing Services Manager on Windows	107
Chapter 12	Installing C2	109
	Deploying C2 Automatically	109
	Installing the C2 Plug-in	109
	Installing C2 Libraries on Microsoft Windows	111
	Installing C2 Libraries on a Linux Environment	111
Chapter 13	Installing O2	113
	Deploying O2 Automatically	113
	Installing the O2 Plugin	113
	Installing O2 Libraries on Microsoft Windows	115
	Installing O2 Libraries on a Linux Environment	115
Chapter 14	Installing D2-Bin	117
	Deploying D2-Bin Automatically	117
	Installing the D2-Bin Plug-in	117
	Installing D2-Bin on Microsoft Windows	119
	Installing D2-Bin on a Linux Environment	119
Chapter 15	Installing D2 RPS Connector	121
	Deploying D2 RPS Connector Automatically	121
	Installing the D2 RPS Connector Plug-in	121
	Installing D2 RPS Connector Libraries on Microsoft Windows	123
	Installing D2 RPS Connector Libraries on a Linux Environment	124
Chapter 16	Installing D2-BOCS	127
	Understanding Documentum Branch Office Caching Servers (BOCS)	127
	Installing the D2-BOCS	127
	Installing D2-BOCS on a BOCS Server for Documentum Server 7.0	129

	Installing D2-BOCS on a BOCS Server for Documentum Server 7.1 or later	130
	Installing D2-BOCS on an ACS server.....	132
	Configuring D2 Client for BOCS.....	133
	Configuring O2 for BOCS.....	134
	Configuring C2 for BOCS.....	137
	Checking D2-BOCS Installation	140
	Enabling BOCS Content Transfer with Non-Anonymous Certificate Based SSL	141
	Enabling Compression for Upload and Download	141
	Enabling Asynchronous BOCS Write	142
	BOCS and ACS Network Locations.....	143
	Setting the BOCS and ACS Network Locations	143
	Using D2-BOCS for Download	143
	Using D2-BOCS for Upload.....	146
Chapter 17	Uninstalling D2	147
	Uninstalling D2.....	147
Chapter 18	Troubleshooting the Installation	149
	D2 Client and D2-Config deployment fails on Jboss server	149
	Unable to launch D2 on Tomcat 8.0.5. Error while fetching repositories.....	149
	Removing old versions of ctx.cab from D2-Config machines	149
	Resolution.....	149
	Disabling xQuery when Using the xPlore Search Engine	150
	Unable to Access D2 Using Microsoft Internet Explorer.....	150
	Configuring Microsoft SQL Server 2008	150
	Unable to Open a New Window or New Connection. No Resource Available.....	151
	DfRegistryWin32.DLL is Already Loaded in Another Classloader	151
	IBM AIX and Apache Tomcat 6.0 Crashing the Server	152
	Files Corrupting During Export.....	152
	D2 Caching and File-Cleaning Services Fail to Operate.....	153
	Null Pointer Exception When Using Reverse-proxy IIS 7 to Import a File Larger Than 25 MB.....	153
	Content Transfer Does Not Go Through the BOCS Server.....	154
	Slow File Transfer When Using a Linux-based Operating System.....	154
	java.lang.ClassNotFoundException when Running an Installer Package.....	155
	No JMS Server Available Exception When Trying Java Method Server (JMS) Failover	156
	InstallException When Installing the Same D2 DAR Files	156
Chapter 19	Configuration Files	157
	Configuration File Locations	157

Preface

OpenText™ Documentum™ D2 consists of two components:

- D2 Configuration: The web-based application, hereafter known as D2 Config, for administrators to use to configure settings such as automated content-handling processes and background settings for D2 Client.
- D2 Client: The web-based application, hereafter known as D2 Client, for users that provides the ability to interact with content in one or more repositories.

When this guide refers to D2, it refers to the product as a whole, not the individual components.

Note: Documentum Content Server is now OpenText Documentum Server. OpenText Documentum Server will be called Documentum Server throughout this guide.

Intended audience

The information in this guide is for system administrators who install and administer Documentum software.

Revision history

The following table lists changes in this guide.

Revision Date	Description
December 2018	Clarified some sections.
April 2018	Initial publication.

Getting Started

Know Before You Start

Before installing, make sure you know:

- How to set CLASSPATH environment variable.
- The install paths for Documentum Server, Java Method Server (JMS), and your web application server.
- How to set variable parameters for the Java Virtual Machine.
- How to modify and deploy a .war package to your application server.
- How to set variable parameters for your application server.

Preparing for Installation

1. Read the *OpenText Documentum D2 Release Notes* for your corresponding version for system requirements.
2. Make sure you have installed:
 - Documentum Server and have configured your repositories and docbrokers.
 - A J2EE web application server as per your enterprise setup.
 - Documentum Composer with a DAR Installer for .dar deployment or Documentum Headless Composer for headless deployment of .dar files.

Note: D2 Installer auto deploys the core D2 .dar files. Documentum Composer with a DAR installer is required for manual deployment of .dar files. [Deploying D2 DAR Files Manually, page 33](#) topic contains further instructions on deploying D2 DAR files manually.
3. Ensure you have administrator privileges on the local system to perform installation.
4. In a Linux environment, set the graphical environment, either by:
 - Adding the variable `java.awt.headless=true` to the environment system properties of the account running the application server.
 - Adding the parameter `-Djava.awt.headless=true` to the Java Method Server startup script.
5. Download the following files to the Content Server and web application server machines:

File	Description	Content
emc-dfs-sdk-x.zip	<p>Documentum Foundation Services (DFS) Software Development Kit (SDK)</p> <p>Download the DFS SDK zip file for the version of Documentum Server you are using.</p> <p>To locate the file, use the search feature for DFS SDK x SPx</p> <p>For example, if you use Documentum Server 7.2, search for DFS SDK 7.2. If you use Documentum Server 7.1 <Patch version>, then get DFS SDK 7.1 <Patch version>. This includes any specific patch level applied to the system.</p>	All content within this archive are needed for installation.
D2_x.zip	Contains the D2 core installer.	D2-Installer-<version>.jar.
All the D2 Starter App files from OpenText My Support .	<p>Contains the sample application to import after installing D2.</p> <p>If you are importing a separate application configuration, you do not need to import the provided sample.</p>	All content is needed for installation.

6. If you are upgrading D2, follow the instructions in [Upgrading D2](#).
If you are upgrading D2 from D2 4.5 or earlier version to D2 16.4, you must run the D2 Migration Utility before upgrading. To run the utility, follow the instructions in [Running the D2 Migration Utility](#), page 18.
7. Follow the instructions for installing D2 in [Instructions for Installing D2](#).

Instructions for Installing D2

The following installation contains the steps for installing the D2-API library files on the Documentum Server, deploying the DAR files, and then installing the D2 Config and D2 Client web applications on the web application server:

1. On the Documentum Server machine:

- a. Extract the contents of `emc-dfs-sdk-x.zip`. Remember the paths to the created folders, because the D2 installation requires you to reference content from the archive.

Note: The `collaboration_services.dar` used by D2 is an updated version of the `.dar` supplied by the DFS SDK. This release of the `.dar` sets the following groups as protected for added security:

- `dce_create_room_groups`
- `dce_datatable_creator`
- `dce_hidden_users`
- `dce_room_creator`
- `dce_user_manager`
- `dcs_privileged_users`

Protected groups can only be used by a client that has been defined as a privileged client.

- b. Follow the instructions in the [Installing D2 on the Content Server](#) section for installing the D2 API libraries. The D2-API is a set of libraries for the Documentum Server and the JMS enabling D2 Methods to be run on the Documentum Server.
 - c. If you are installing D2 on a Content Server cluster, install the D2 API libraries on each instance of Documentum Server.
2. On the web application server:
 - a. Stop web application server services.
 - b. Follow the instructions for installing D2 on the web application server as described in the following table:

Web application server	Instructions
Apache Tomcat for Microsoft Windows	Installing D2 on Apache Tomcat for Microsoft Windows
Apache Tomcat for a Linux environment	Installing D2 on Apache Tomcat for a Linux Environment
IBM WebSphere	Installing D2 on IBM WebSphere
Oracle WebLogic	Installing D2 on Oracle WebLogic
Redhat JBOSS	Installing D2 on Redhat JBOSS
VMware vFabric tcServer for Microsoft Windows	Installing D2 on VMware vFabric tcServer for Microsoft Windows

- c. Start application server services.
3. Configure D2:

The D2 installer handles common Documentum Server and application server configuration options, these settings may be changed or additional optional settings may be configured post-installation.

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

Configuration	Instructions
Configuring <code>logback.xml</code>	Configuring logback.xml for the Content Server
Configuring the display of tables	Configuring Content Server Table Display
Configuring D2 auditing	Configuring D2 Auditing

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

Application	Instructions
D2 Config	Configuring D2 Config
D2 Client	Configuring D2 Client
D2 Java Method Server	Configuring D2 Java Method Server

4. Configuring authentication protocols if you want to use Windows NT Unified Logon (NTLM) or Kerberos as described in the following table:

Authentication	Instructions
NTLM	Configuring NT Unified Logon (NTLM)
Kerberos	Configuring Kerberos
TrustedReverseProxy	Configuring TrustedReverseProxy for Various SSO Environments

5. You can configure the file transfer mode. [File Transfer Modes](#) contains more information about the Java and thin client file transfer modes and [Configuring File Transfer Modes](#) contains configuration instructions.
6. Run the applications.
7. Log in to D2 Config and import configurations. The *OpenText Documentum D2 Administration Guide* contains further instructions on importing configurations.
- Using D2 without configurations can cause errors. If you do not have a previous set of configurations to import, download and import **HR Config D2 4.x - Export-Config.zip** as a sample set of configurations.

Understanding the D2 Keystore Utility

The D2 Keystore utility replaces the functionality previously provided by the D2 Lockbox utility. If you are upgrading from a version of D2 where Lockbox is installed, it is not necessary to uninstall Lockbox, but instructions for removing Lockbox artifacts manually are provided in [Removing Lockbox Artifacts](#), page 26.

Use the D2 Keystore utility (**D2KeystoreUtil**) to read from and write to the singleton `d2_keystore` object in the global registry repository.

Note: You must deploy **D2-DAR.dar** to the global registry repository before the D2 Keystore utility can be run, or before any D2 DFC Client (such as the D2 Config or D2 Client web applications) can be run.

By default, the d2 keystore holds two encryption keys:

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

The values for these properties are automatically generated by D2 and there is no need to enter or modify these quantities using the D2 Keystore utility unless it is specifically desired to change these for whatever reason. Note that if the value of the `encryption_key` property is changed, then one needs to re-enter the `d2_mail_config` password property values for each relevant repository using D2-Config (see *Configuring the Mail Server* in the *D2 Administration Guide*).

The d2 keystore is also used to hold various sensitive quantities such as repository admin login names and passwords that are needed for various optional features of D2 including Single-Sign On (SSO) and LoadOnStartup initialization of dictionary caches, etc.

When the utility reads, it creates or overwrites the `d2keystore.properties` file in the current directory with the current contents of the d2 keystore. When the utility writes, it clears the d2 keystore in the global registry repository, reads the `d2keystore.properties` file in the current directory, writes these properties to the d2 keystore in the global registry repository, and then deletes the `d2keystore.properties` file from the current directory.

Note: If the read command (`-r`) is run without running the write (`-w`) command, the `d2keystore.properties` file remains on the system. This file contains sensitive credential information and should be deleted manually if there are no plans to use the d2keystore utility to write information back into the d2keystore repository object.

Note: The global registry must be configured correctly with the proper password in `dfc.properties`.

Running the D2 Keystore Utility

Note: You must run the D2 Keystore utility whenever a password changes for one of the administrative accounts that you have previously stored in the d2 keystore.

1. In a command shell, navigate to the directory in which the D2-Config web application has been deployed, and then to the `d2keystore` subdirectory. For example, on Windows, this might be `C:\Tomcat\webapps\D2-Config\utils\d2keystore`, while on Linux, this might be `/opt/Tomcat/webapps/D2-Config/utils/d2keystore`.
2. Run the following command to read the current d2 keystore properties and write them to a `d2keystore.properties` file. Note that the credentials you provide to the utility must be those of a superuser account in the global registry repository.
 - On Windows: `.\D2KeyStoreUtil.cmd -u superusername -p password -r`
 - On Linux: `./D2KeyStoreUtil.sh -u superusername -p password -r`
3. Edit the file `d2keystore.properties` and add appropriate lines for the properties that you wish to set in the d2 keystore.
 - If you are configuring D2 SSO using DFC principal mode authentication. Add the following two lines to specify admin credentials for all repositories of interest:

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

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

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

- If you are configuring D2 app server for LoadOnStartup initialization, you would add the following two lines to specify admin credentials for all repositories of interest:

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

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

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

4. Run the following command to read the values from d2keystore.properties and write them to the d2 keystore:

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

Note: If either the `transient_encryption_key` or `encryption_key properties` are not found in the `d2keystore.properties` file produced by running `D2KeyStoreUtil -r`, they will be automatically generated and written to the D2 keystore when the utility is run again with the `-w` option.

Note: To get usage information for either of these scripts, run them without arguments. The current version of the underlying java command line utility outputs:

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

```
-r : Read d2_keystore properties from global registry repository and write
to ./d2keystore.properties. The d2_keystore properties file will remain on
the system until the -w command is given or it is manually removed.
```

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

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

All command line argument switches are case-sensitive.

Note: To run these scripts and the underlying java command line utility from a different directory, copy the scripts and the `D2KeyStoreUtil.jar` file to this directory and edit the `webinfdir` variable inside the relevant script accordingly. The value of this variable must be the path name (relative or absolute) for the WEB-INF directory of a deployed D2-Config web application.

Note that only one repository can be designated as the global registry repository. If this designation changes, careful planning is required in order to migrate the D2 Keystore accordingly:

1. Before changing the designation, export the D2 Keystore to a `d2keystore.properties` file using the D2 Keystore utility with the `-r` option.
2. After changing the designation, import the `d2keystore.properties` created in step 1 using the D2 Keystore utility with the `-w` option.

Note: if the global registry repository changes, the `dfc.globalregistry.repository` must be changed in the `dfc.properties` files for all relevant DFC clients, including the `dfc.properties` file used by the D2 Keystore command line utility.

Registering D2 DFC Clients as Privileged Clients

All D2 DFC Clients (D2 servers, D2-Config servers, D2 REST servers, D2-BOCS/BOCS servers or BPM servers to which D2 has been deployed) must be approved as privileged clients in each applicable repository. It is not necessary to approve the Documentum Java Method Server (JMS) DFC client as a privileged client because it runs on the Documentum Server machine and is automatically a Trusted DFC Client. Failure to approve a D2 DFC Client as a privileged client will result in errors when that DFC Client attempts to access the D2 keystore in the global registry repository or attempts to execute code in privileged mode.

Approval is granted using Documentum Administrator (DA).

1. You can view the list of DFC clients for a given repository that can be approved as privileged clients in DA by navigating to **Administration > Client Rights Management > Privileged Clients**. See the *Privileged Clients* section in the *OpenText Documentum Server Administration and Configuration Guide* for more information.
2. If the clients that you want to approve as a privileged client do not appear in the list, click **Manage Clients**. A list of all registered DFC clients known to the repository is displayed.

Note: Note that a DFC client is registered the first time it connects to the Documentum Server. You might need to run the client once (for example, run the D2-Config web application) in order to register it. To help you locate clients in the list of registered clients, you can filter the list by Client Name by entering a prefix in the text box with help text "**Starts with**" and clicking the arrow button to the right.

3. Select the desired clients from the list on the left-hand side and insert them into the list on the right-hand side. Click **OK**.
4. Right-click on a client and select **Approve Privilege**.

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

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

To confirm that the client you selected is the one that you want to approve as a privileged client, you might need to compare its Client ID value as displayed in DA with the client ID value stored in its **dfc.keystore** file.

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

- a. Open a command prompt and navigate to the folder where the **dfc.properties** file for your DFC client is located. Typically, this folder is the `WEB-INF/classes` folder of the associated web application. If the **dfc.security.keystore.file** property in the **dfc.properties** file has not been set, then this folder contains the desired **dfc.keystore** file. If it has been set, then the value of this property will give the path to the desired **dfc.keystore** file. Note that the **dfc.keystore** file is created the first time a DFC client connects to the Documentum Server.
- b. Once you have located the folder for the **dfc.keystore** file, navigate to this folder in a command shell and run the java keytool command:

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

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

```
Owner: CN=dfc_t2zN2bJv8DDyMlZbXD9qJjTWiXAa, O=EMC, OU=Documentum
```

The desired Client ID is `dfc_t2zN2bJv8DDyMlZbXD9qJjTWiXAa`.

Upgrading D2

Planning Your D2 Upgrades

The following table lists the upgrade tasks required to upgrade D2 (read top to bottom), based on your current D2 version:

Note: All the D2 versions must be first upgraded to D2 4.5 before running the D2 Migration Utility and before upgrading to D2 16.4.

Upgrade Tasks	D2 4.0	D2 4.1	D2 4.2	D2 4.5	D2 4.6	D2 4.7
Upgrade D2 to D2 4.2	Required [1]	N/A	N/A	N/A	N/A	N/A
Upgrade Documentum Server to supported 7.x version	N/A	CS 7.0, 7.1 [4]	CS 7.0, 7.1, 7.2 [5]	CS 7.0, 7.1, 7.2, 7.3	CS 7.1, 7.2, 7.3	CS 7.1 and higher
Upgrade D2 to D2 .4.5	See note [2]	Required	Required	N/A	N/A	N/A
Run D2 Migration Utility [3]	Required	Required	Required	Required	N/A	N/A
Upgrade D2 to D2 16.4	Required	Required	Required	Required	Required	Required

Note: [1] D2 4.0 is not supported on a 7.x platform. As such, D2 4.0 must first be upgraded to a version of D2 that is supported on the currently underlying Documentum Server platform. If current Documentum Server platform is pre 6.7 SP2 then upgrade to D2 4.2, else upgrade to D2 4.5.

[2] If D2 is not yet at version 4.5, then this step is required.

[3] D2 Migration is an irreversible process. Restore to pre-migration state is possible only through database backup, restore, or any other snapshot type restore. D2 Migration Utility does not impact files on the file system, it only affects D2 Config objects in the repository. To run the utility, follow the instructions in [Running the D2 Migration Utility, page 18](#)

[4] Documentum Server 7.1 is supported with D2 4.1 P18 and higher.

[5] Documentum Server 7.2 is supported with D2 4.2 P17 and higher.

Pre-requisites:

1. Delete existing user preferences with the DQL query `delete d2c_preferences objects;`
2. Stop web application services.

Note: The DocBroker services must be running.

3. Delete the cookies and cache of the web browsers.
4. Back up the previous WAR files. [Locations of Configuration Files](#) contains the locations of configuration files.
5. If you are upgrading from D2 4.2 or 4.5 and POI files are present in the JMS libraries, remove all `poi-<type>-3.6.jar` files from the JMS libraries located as described in the following table and retain all `poi-<type>-3.9.jar` files. The presence of the 3.6 files after upgrading may prevent D2 and the O2 plug-in from functioning correctly.

Documentum Server 7.1 and later	Documentum Server 7.0
<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/ServerApps.ear/lib</code>	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/ServerApps.ear/lib</code>

6. Delete the temporary installer files. For example, in Microsoft Windows delete the folder `C:\Users\Administrator\AppData\Local\Temp\2\D2-Installer_<version>`

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

Note: Make sure you only delete the contents of the `%USERPROFILE%\AppData\Local\Temp\#` folder and not the folder itself. The JVM API uses this folder to store the temporary files. If this folder is deleted, the JVM API will reference a non-existent location that will result into errors when running the D2 core installation and D2 configuration utility. If `%USERPROFILE%\AppData\Local\Temp\#` folder is deleted, restart the server before running the D2 core installation and D2 configuration utility.

Running the D2 Migration Utility

Note: Running the migration utility is only required when you are using D2 version 4.5 or earlier and planning to upgrade to D2 16.4. If you are already using D2 4.6 or later, the migration utility is not needed.

If you are planning to upgrade your D2 version, check for the pre-requisites and upgrade tasks required to upgrade D2.

The D2 Migration Utility is a DFC standalone utility that is run on the Documentum Server machine to migrate the D2 config objects in the docbase.

The D2 Migration Utility must be run to ensure existing D2 environments adhere to the new config object model which now extends `dm_sysobject` and thereby grants appropriate ACLs to config objects. The utility is shipped as a standalone zip file along with the required libraries.

The D2 Migration Utility works in the following phases:

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

Remember the following points before running the D2 Migration Utility:

- The migration utility requires that **DM_HOME** environment variable is set on the machine. This is required to invoke the Headless Composer pre-packaged along with the docbase. This is set by default.
- “Error retrieving object by Object Id. This may happen if an object previously installed by Composer was deleted” and DM_TYPE_MGR_E_CANT_FIND_TYPE_HANDLE in composer logs will be encountered while running the migration utility. This does not impact the migration in any manner and can be safely ignored.
- Run the migration utility from the Documentum Server machine where migration is performed.
- In Windows Server 2012, run command prompt as administrator if the windows logged in user does not have administrator privileges.
- The docbase user should be a superuser.
- Install Java 1.7 or later on the Documentum Server machine where the utility runs.
- Migration phases can be controlled by command line arguments.
- The migration utility can be re-run if migration fails at any point in time.
- The migration utility performs migration by doing dump, deletion, and recreation of data. Hence, it is strongly advised to backup the Documentum Server and Database machines before performing migration in production environment. The process cannot be reverted except via backup restores or snapshot based solutions.
- In D2 version 4.2, the **Dictionary** attribute in creation profiles was not mandatory, but it is mandatory starting in version 4.5. Set the **Dictionary** drop-down before migrating to avoid corrupting the **creation_profile**.
- To avoid possible data corruption issues, consult with OpenText Support before manually terminating the migration utility.

To run the D2 Migration Utility, perform the following steps:

1. Extract D2-Config-Migrator-16.4.0.zip to the Documentum Server machine.
2. The dfc.properties file in the folder by default, includes C:/Documentum/config/dfc.properties. For non-Windows machines, update the path accordingly.
3. Check Services:
 - a. Ensure that the Documentum Server and docbroker services are running.

- b. Stop the web application server and JMS.
4. Open command prompt and run the utility using following command:

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

Set the parameters for the command as described in the following table:

Parameter	Description
<docbase>	Name of the docbase repository.
<loginname>	Install owner login name.
<password>	Install owner password.
skipCreateConfigFolders	(Optional) Skips creating sub-folders for every config type under /System/D2/Data in repository.
skipDumpConfigs	(Optional) Skips dumping D2 config objects
skipScanDumpRelations	(Optional) Skips dumping dm_relations related to these configs.
skipDumpTypes	(Optional) Skips dumping dm_type that are migrated.
skipScanDumpedConfigs	(Optional) Skips scanning dumped config xml files to determine objects already migrated in previous run.
skipScanDumpedRelations	(Optional) Skips scanning dumped relation xml files to determine relations already migrated in previous run.
skipDeleteConfigs	(Optional) Skips deleting config objects from repository.
skipDeleteTypes	(Optional) Skip dropping types from repository.
skipInstallDars	(Optional) Skips headless composer dar upgradation.
skipCreateConfigs	(Optional) Skips recreation of configs from dump files.
skipCreateRelations	(Optional) Skips recreation of relations from dump files.
skipUpdateRelationDescriptions	(Optional) Skips updating object id references in dm_relation objects.
forceMigrate	(Optional) Force migration on the repo. Dump config, dar installation, and delete types phases will be forced irrespective of arguments.
help	(Optional) Lists out all possible arguments for the command.

The optional arguments need not be used unless the migration fails repeatedly and an override is required for any of the phases.

5. The Migration Utility does not register the table for a dictionary you selected for **Create Register Table**. In this case, follow these manual registration steps:
 - a. In D2 Config, navigate to the applicable dictionary and deselect **Register Table**.
 - b. Save the dictionary, then select it again and save a second time. A register table will be created.
 - c. To confirm the creation of the register table, run the following DQL query and review the result: **select * from dbo.dictionaryname**
6. Validate the migration.

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

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

Example 2-1.

Upgrading D2 4.6 to D2 16.4 on the Documentum Server

Before upgrading ensure that the following prerequisites are met:

- If they are present in your Documentum Server CLASSPATH, remove `C6-Common <version number>.jar`, `D2.jar` references.
- Remove the WAR files created by the previous installation
- Navigate to the `C:\Documentum` folder and remove the `.D2InstallationInfo` files. If the `.D2InstallationInfo` file is present it will only allow for an express or custom upgrade and not prompt for the correct versions or locations of plugins and the Documentum Server JMS path.
- If you are using JBoss or Wildfly to deploy D2 you will have to update the java policy or `permissions.xml` first, depending on your version. If you are using JBoss 6.x, update the JRE by editing or creating the `java.policy (/lib/security/java.policy/java.policy)` file with the following permission:

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

1. Launch D2 Installer and click **Next**.

Note: If you do not have administrator rights, select **Run installer as administrator**.

- **For Microsoft Windows:**

Right-click on **D2-Installer-<version>.jar**, select **Open with**, and then select **Java(TM) Platform SE binary**.

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

```
C:\Document and Settings\\Local Settings\Temp\D2-Installer_x  
[user] is the user name of the account, and x is the version number.
```

The temporary directory holds the installation logs.

- **For Linux Environment:**

Open an xterm and run the installer by typing `java -jar D2-Installer-<version>.jar`.

The installation logs will be written into `c:\User\`.

Note: In previous versions of D2 that used Lockbox, the version of Java the install utility used was important. With the removal of Lockbox it is no longer necessary to ensure 32/64bit Java is used, but after installing D2 please ensure you grant the D2 applications 'Privileged DFC client' access in your D2 repositories. Please refer to *Privileged Clients* in the *OpenText Documentum Server Administration and Configuration Guide* for instructions.

2. Select the **Install D2** options to start the installation of D2 on the Documentum Server . Click **Next**.
3. On the **Select installation packages** page, select **D2, D2-Config, D2-API for Content Server\JMS, D2-API for BPM, and DAR**. Click **Next**.

Note: To install D2 APIs for BPM, you must have Documentum Process Engine installed on your machine.

4. On the **WebApp extraction folder** page, select a folder to which the installer extracts **D2.war** and **D2-Config.war**. The selected folder must not already contain these war files.

If you are using a JBoss 5.x web application server, select **Remove xerces libraries (mandatory for jboss 5.X)** to remove a file that conflicts with the JBoss web application server. If you are not using a JBoss 5.x web application server, do not select the checkbox. Selecting it causes the installer removes files required for D2. Click **Next**.

5. On the **D2 plugins installation** page, click **Add a plugin** and add the installer .jar files to install new plug-ins or update existing plug-ins. Click **Next**.

You can add AdvancedPublishing, C2, O2, D2-Bin, and D2-RPS Connector plug-ins. The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, and automatically deploys DAR files. For example, if you add **C2-Install.jar**:

- You do not need to deploy the output **C2-API.jar** and **C2-Plugin.jar** files.
- You do not need to configure **D2-Config.properties**.
- You do not need to deploy **C2-DAR.dar**.

If you do not add plug-ins using the D2 Installer, you can manually run the plug-in installer and deploy the .jar files after completing the D2 installation.

6. Fill out the **D2-API extraction folders** page as described in the following table:

Field	Path
For Documentum Server	<install path of D2>
For Java Method Server	<p>For Documentum Server Version 7.1 and later, use <install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deployments\ServerApps.ear</p> <p>For Documentum Server Version 7.0, use <install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\ServerApps.ear</p>
For Business Process Manager	<p>For Documentum Server Version 7.1 and later, use <install path of Documentum>\<Java Method Server>\DctmServer_MethodServer\deployments\bpm.ear</p> <p>For Documentum Server Version 7.0, use <install path of Documentum>\<Java Method Server>\DctmServer_MethodServer\deploy\bpm.ear</p>

7. On the **DAR extraction folder** page, select the path to extract the DAR files. Click **Next**.

This step produces the **D2-dar.dar**, **D2Widget-dar.dar**, and **Collaboration_Services.dar** files in the selected folder.

Note: The D2 installer automatically deploys the DAR files as part of the installation process.

8. On the **Documentum dependencies** page, for **Path**, locate and select the folder to which you extracted the DFS SDK. Click **Next**.

Note: Make sure you select the main DFS SDK folder that contains the **lib** folder. Ensure that you are using the same version of Documentum Server. For example, use DFS SDK 7.1 for Documentum Server 7.1.

The installer automatically includes necessary files in the extracted WAR files. You can delete the DFS SDK library files after the D2 installation completes.

9. On the **User Data** pages, perform the following:

- a. Type the Documentum Server install owner's name and password. Click **Next**.
- b. Type the repositories name (separated by comma) for which you would like to install D2. D2 will be deployed on each of the repositories provided here.
- c. Select **Yes** to prevent repeating attributes from being returned as individual rows in lists such as advanced searches, property pages, and repository browser widgets. Click **Next**.

- d. Select **Yes** to force D2 to apply Autolink rules before applying Security rules to the content. Select **No** to force D2 to apply Security rules to the content before applying Autolink. Click **Next**.
10. Once the DAR installation is finished, click **Next**, then click **Done**.

Re-running the Upgrade Utility

Once D2 16.4 has been installed, the details of the installation is saved in the `.D2InstallationInfo` file in the <Documentum installation folder>. For example, `C:\Documentum`.

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

1. Launch D2 Installer and click **Next**.
2. Select **Upgrade D2**.
You get two upgrade options:
 - Express Upgrade
 - Customized Upgrade
3. Select **Express Upgrade** to upgrade your D2 installation using the previous installation settings. Click **Next** and perform the following.
 - a. Review the summary page for all the parameters and pre-configured values. Click **Next**.
 - b. On the **D2 plugins installation** page, click **Add a plugin** and add the installer `.jar` files to install new plug-ins or update existing plug-ins. Click **Next**.
You can add Advanced Publishing, C2, O2, D2-Bin, and D2-RPS Connector plug-ins. The D2 Installer automatically runs the added plug-in installer `.jar` files and activates the plug-in for D2, and automatically deploys DAR files. For example, if you add **C2-Install.jar**:
 - You do not need to deploy the output **C2-API.jar** and **C2-Plugin.jar** files.
 - You do not need to configure **D2-Config.properties**.
 - You do not need to deploy **C2-DAR.dar**.If you do not add plug-ins using the D2 Installer, you can manually run the plug-in installer and deploy the `.jar` files after completing the D2 installation.
 - c. Type the Documentum Server install owner's name and password. Confirm the password and click **Next**.
4. Select **Customized Upgrade** to proceed with a full D2 installation. During customized installation, all the values from the previous successful installation will be populated automatically. The values can be modified if required. Follow the instructions for upgrading D2 in [Upgrading D2 4.6 to D2 16.4 on the Documentum Server, page 21](#).

Upgrading D2 on the Web Application Server

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

- Application server services are stopped before running the installer.
- Unlike versions of D2 before 4.6, do not extract the new D2 .war files before running the D2 installer on the application server.

Upgrading D2 on Application Server includes the process of updating the D2 configuration files. This process reads existing configuration files from D2 and D2-Config web application servers and updates the same files within the new D2 and D2-Config .war files. These configuration files include:

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

Note: As of D2 version 4.7, D2FS-trust.properties was made obsolete and has been replaced by the D2 keystore.

Note: The names of configuration files are case-sensitive.

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

To update Application Server D2 configuration files:

1. Copy **D2-Config.war** and **D2.war** to the **<install path of the web application server>\webapps** folder.
2. Clear the cache and the temporary folder.
 - a. On Apache Tomcat for a Linux environment:
 - a. Clear the Catalina cache in the folder **<install path of Tomcat>\work\Catalina\localhost**
 - b. Clear the Tomcat temporary folder **<install path of Tomcat>\temp**
 - b. On Redhat JBoss, clear the temp folder **<install path of the web application server>/standalone/tmp.**
 - c. On VMware vFabric tcServer for Microsoft Windows:
 - a. Clear the cache folder **<install path to Tc-server>\logs**
 - b. Clear the temp folder **<install path to Tc-server>\temp**
 - c. Clear the work folder **<install path to Tc-server>\work**

3. Launch D2 Installer and click **Next**.

Note: In previous versions of D2 that used Lockbox, the version of Java the install utility used was important. With the removal of Lockbox it is no longer necessary to ensure 32/64bit Java is used, but after installing D2 please ensure you grant the D2 applications 'Privileged DFC client' access in your D2 repositories. Please refer to *Privileged Clients* in the *OpenText Documentum Server Administration and Configuration Guide* for instructions.

4. Select **Update Application Server D2 Configuration Files**. Click **Next**.
5. Select an application server from the list. Click **Next**.
6. Select a web application server you like to configure from the list. Click **Next**.
The list includes Apache Tomcat, IBM WebSphere, Oracle WebLogic, Redhat JBOSS EAP, and VMware vFabric tcServer
7. Specify the location of the old webapps folder **<install path of the web application server>\webapps** where the old configurations files are located. Click **Next**.
8. Specify the location where new D2.war and D2-Config.war are located. Click **Next**.
9. (Optional) Select **Yes** to set the load on startup information. Click **Next**. Use load on startup to pre-cache the D2 configuration information. This results in better initial performance for users when a system has been restarted but results in a longer startup time for the application server.
 - a. Type the repository name, user name, domain, and install owner's password for the repository that you want to set for load on startup.
 - b. Select **Yes** to set load on startup information for more repositories. Click **Next**.
 - c. Type the repository name, user name, domain, and install owner's password for each repository. Click **Next**.
Note: You can input information for at most five repositories.
10. Read the extraction summary and click **Next**, then click **Done**.
11. Start the Application Server.

Example 2-2.

Removing Lockbox Artifacts

Lockbox support has been removed from D2 as of version 4.7. If you are upgrading from pre 4.7 D2, it is recommended (but optional) to remove Lockbox elements from your environment manually. Credentials are now stored in the D2 keystore. For more information about moving this information to keystore, see [Understanding the D2 Keystore Utility, page 12](#).

1. Complete the upgrade from D2 4.6 or earlier to D2 16.4.
Note: During the initial installation and configuration of Lockbox, you chose directories on your Documentum Server, ACS server, BOCS server and Application servers where related files would be stored. These files are identified as **D2.lockbox**, and can be deleted. **LB.jar** and **LBJNI.jar** files can also be deleted if they are still present. See below for details on each context.
2. Remove Lockbox items from your Documentum Server:

- a. Remove the Lockbox files directory and its sub-folders and files. This folder (~lockbox_directory) generally contains the D2.lockbox and backup files. It also contains a folder called ~lockbox_directory\lib which contains **C6-Common.jar**, **LB.jar** and **LBJIN.jar** files. The ~lib folder contains another folder called ~lockbox_directory\lib\native. The native directory contains platform specific libraries. For example, for windows it is **win_vc100_x64** if you are using 64-bit.
- b. Go to ~<DOCUMENTUM_HOME>\<Java Method Server>\modules\emc\ and remove the folder called 'd2' and its sub-folder and files. The folder contains the following sub-folder ~<DOCUMENTUM_HOME>\<Java Method Server>\modules\emc\d2\lockbox\main and files such as **LB.jar** , **LB.jar.index**, **LBJNI.jar**, **LBJNI.jar.index** and **modules.xml**
- c. Go to ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\ServerApps.ear \APP-INF\classes and remove D2.lockbox and its backup files.
- d. Open the **jboss-deployment-structure.xml** from the location ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\ServerApps.ear\META-INF\ and remove the following section, present under the **<deployment>** tag, then save the file.

```
<dependencies>
  <module name="emc.d2.lockbox"/>
</dependencies>
```
- e. If it exists, please remove ~<Documentum Home>\D2\Lockbox and its sub-folder. It generally contains **LB.jar** and **LBJNI.jar** and platform dependent library files.
- f. Remove references to **C6-Common.jar**, **LB.jar**, **LBJNI.jar** from \$CLASSPATH.
- g. Remove references to platform related library files from the PATH variable. For example, **win_vc100_x64** if you are using Windows 64-bit.

3. Remove Lockbox items from BPM:

- a. Go to ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\bpm.ear\APP-INF\classes and Remove the D2.lockbox and its backup files.
- b. Go to ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\bpm.ear\META-INF\ Open the **jboss-deployment-structure.xml** file remove the below entry available under the **<dependencies>** tag and save the file:

```
<module name="emc.d2.lockbox"/>
```

4. Remove Lockbox items from ACS:

- a. If applicable, remove the D2.lockbox file from ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\acs.ear\D2-BOCS.war\WEB-INF\classes folder.
- b. Go to ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\acs.ear\META-INF and open the **jboss-deployment-structure.xml**. Remove **<dependencies>** tag present under the tag **<sub-deployment name="D2-BOCS.war">** and save the file:

```
<dependencies>
```

```
<module name="emc.d2.lockbox"/>
```

```
</dependencies>
</sub-deployment>
```

5. Remove Lockbox items from BOCS:
 - a. Remove the Lockbox files directory and its sub-folders and files. This folder (~lockbox_directory) generally contains the D2.lockbox and backup files. It also contains a folder called ~lockbox_directory\lib which contains **C6-Common.jar**, **LB.jar** and **LBJIN.jar** files. The ~lib folder contains another folder called ~lockbox_directory\lib\native. The native directory contains platform specific libraries. For example,; for windows it is **win_vc100_x64** if you are using 64-bit.
 - b. If applicable, remove the D2.lockbox file from the ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_BOCS\deployments\D2-BOCS.war\WEB-INF\classes folder
 - c. If applicable, remove **LB.jar** and **LBJNI.jar** from ~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_BOCS\deployments\D2-BOCS.war\WEB-INF\lib folder
 - d. Remove references to **C6-Common.jar**, **LB.jar**, **LBJNI.jar** from \$CLASSPATH.
 - e. Remove references to platform related library files from the PATH variable. For example, **win_vc100_x64** if you are using Windows 64-bit.
6. Remove Lockbox items from your Application Server:
 - a. Remove the Lockbox files directory and its sub-folders and files. This folder (~lockbox_directory) generally contains the D2.lockbox and backup files. It also contains a folder called ~lockbox_directory\lib which contains **C6-Common.jar**, **LB.jar** and **LBJIN.jar** files. The ~lib folder contains another folder called ~lockbox_directory\lib\native. The native directory contains platform specific libraries. For example,; for windows it is **win_vc100_x64** if you are using 64-bit.
 - b. Go to the Application Server \lib folder and remove the **LB.jar** and **LBJNI.jar** files.
 - c. Remove references to **C6-Common.jar**, **LB.jar**, **LBJNI.jar** from \$CLASSPATH.
 - d. Remove references to platform related library files from the PATH variable. For example, **win_vc100_x64** if you are using Windows 64-bit.
 - e. If applicable, remove the parameter **lockboxPath** from D2FS/D2-Config.Properties files. When installing D2 16.4, the parameter **lockboxPath** will not be available unless you manually copy the D2FS/D2-Config.Properties from the previous D2 versions.
7. Delete references to **SetLockboxProperty** in **D2-Config.properties**.

Installing D2 on the Documentum Server

Installing D2 on the Documentum Server

The following installation contains the steps for installing D2 on the Documentum Server.

Note: If you are using JBoss or Wildfly to deploy D2 you will have to update the java policy or permissions.xml first, depending on your version. If you are using JBoss 6.x, update the JRE by editing or creating the `java.policy (/lib/security/java.policy/java.policy)` file with the following permission:

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

If you are using JBoss 7.x or Wildfly 9.x, add the permissions.xml in `ServerApps.ear\META-INF` and `BPM.ear\META-INF` on the Documentum Server to give propagate action for RolePermission:

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

1. Launch D2 Installer and click **Next**.

Note: If you do not have administrator rights, select **Run installer as administrator**.

Note: Execute any permissions in the default temp directory of the host.

- **For Microsoft Windows:**

Right-click on `D2-Installer-<version>.jar`, select **Open with**, and then select **Java(TM) Platform SE binary**.

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

`C:\Users\<<username>\Local Settings\Temp\D2-Installer_x`

[user] is the user name of the account, and x is the version number.

The temporary directory holds the installation logs.

- **For Linux Environment:**

Open an xterm and run the installer by typing `java -jar D2-Installer-<version>.jar`.

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

```
/tmp/D2-Installer_x
```

The temporary directory holds the installation logs.

Note: In previous versions of D2 that used Lockbox, the version of Java the install utility used was important. With the removal of Lockbox it is no longer necessary to ensure 32/64bit Java is used, but after installing D2 please ensure you grant the D2 applications 'Privileged DFC client' access in your D2 repositories. Please refer to *Privileged Clients* in the *OpenText Documentum Server Administration and Configuration Guide* for instructions.

2. Select **Install D2** to start the installation of D2 on the Documentum Server . Click **Next**.
3. On the **Select installation packages** page, select **D2, D2-Config, D2-API for Content Server\JMS, D2-API for BPM, and DAR**. Click **Next**.

Note: To install D2 APIs for BPM, you must have Documentum Process Engine installed on your machine.

4. On the **WebApp extraction folder** page, select a folder to which the installer extracts **D2.war** and **D2-Config.war**. The selected folder must not already contain these war files.

If you are using a JBoss 5.x web application server, select **Remove xerces libraries (mandatory for jboss 5.X)** to remove a file that conflicts with the JBoss web application server. If you are not using a JBoss 5.x web application server, do not select the checkbox. Selecting it causes the installer removes files required for D2. Click **Next**.

5. On the **D2 plugins installation** page, click **Add a plugin** and add the installer .jar files to install new plug-ins or update existing plug-ins. Click **Next**.

You can add AdvancedPublishing, C2, O2, D2-Bin, and D2-RPS Connector plug-ins. The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, and automatically deploys DAR files. For example, if you add **C2-Install.jar**:

- You do not need to deploy the output **C2-API.jar** and **C2-Plugin.jar** files.
- You do not need to configure **D2-Config.properties**.
- You do not need to deploy **C2-DAR.dar**.

If you do not add plug-ins using the D2 Installer, you can manually run the plug-in installer and deploy the .jar files after completing the D2 installation.

6. Fill out the **D2-API extraction folders** page as described in the following table:

Field	Path
For Documentum Server	<install path of D2>
For Java Method Server	For Documentum Server Version 7.1 and later, use <install path of Documentum>\<Java method server>\server\DctmServer_MethodServer\deployments\ServerApps.ear For Documentum Server Version 7.0, use <install path of Documentum>\<Java method server>\server\DctmServer_MethodServer\deploy\ServerApps.ear
For Business Process Manager	For Documentum Server Version 7.1 and later, use <install path of Documentum>\<Java method server>\DctmServer_MethodServer\deployments\bpm.ear For Documentum Server Version 7.0, use <install path of Documentum>\<Java method server>\DctmServer_MethodServer\deploy\bpm.ear

7. On the **DAR extraction folder** page, select the path to extract the DAR files. Click **Next**.

This step produces the **D2-dar.dar**, **D2Widget-dar.dar**, and **Collaboration_Services.dar** files in the selected folder.

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

8. On the **Documentum dependencies** page, for **Path**, locate and select the folder to which you extracted the DFS SDK. Click **Next**.

Note: Make sure you select the main DFS SDK folder that contains the **lib** folder. Ensure that you are using the same version of Documentum Server. For example, use DFS SDK 7.1 for Documentum Server 7.1.

The installer automatically includes necessary files in the extracted WAR files. You can delete the DFS SDK library files after the D2 installation completes.

9. On the **User Data** pages, perform the following:
- Type the Documentum Server install owner's name and password. Click **Next**.
 - Type the repositories name (separated by comma) for which you would like to install D2. D2 will be deployed on each of the repositories provided here.
 - Select **Yes** to prevent repeating attributes from being returned as individual rows in lists such as advanced searches, property pages, and repository browser widgets. Click **Next**.

- d. Select **Yes** to force D2 to apply Autolink rules before applying Security rules to the content. Select **No** to force D2 to apply Security rules to the content before applying Autolink. Click **Next**.
10. Once the DAR installation is finished, click **Next**, then click **Done**.

Deploying D2 DAR Files Manually

Deploying D2 DAR Files Manually

Note: D2 Core installer automatically deploys the following DAR files. If needed, these DAR files can also be manually deployed as described in the table.

DAR	Description
D2-DAR.dar	Deploy to install the core D2 data model to a repository. Contrary to previous versions of D2, it is now necessary to deploy D2-DAR.dar to the global registry repository. This will also cause themes to be properly and consistently displayed across all D2 repositories.
D2Widget-DAR.dar	Deploy to install D2 Client.
Collaboration_Services.dar	Deploy to use the Comments widget in D2 Client. If you are not hosting the D2 web application in the global repository, you must deploy Collaboration_Services.dar to both the global repository and the repository hosting the D2 web application. Use a 6.7 SP2 or later version of dardeployer or headless composer when deploying Collaboration_Services.dar .
Plug-in DAR files, such as C2-DAR.dar .	Deploy to use the respective plug-ins. AdvancedPublishing, C2, O2, D2-Bin, and D2-RPS Connector dar files are deployed for the respective plug-ins if indicated during the D2 installation process.

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

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

Field	Description
DAR	Locate and select the DAR file.
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the nodmadmin.installparam.xml file if the Documentum Server installation owner is not named dmadmin, as described in Step 3.

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


```
<?xml version="1.0" encoding="UTF-8"?>
<installparam:InputFile xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:installparam="installparam">
<parameter key="dmadmin" value="<administrator account name>" />
</installparam:InputFile>
```
 - c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the **nodmadmin.installparam.xml** file you created.
4. Click **Install**.
5. Click **Recent DAR install log files** to review log files.
6. Return to the instructions: [Instructions for Installing D2](#).

Installing D2 on the Web Application Server

Limitations of D2 Core Installer

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

Installing D2 on Apache Tomcat for Microsoft Windows

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

To update the references:

- a. Navigate to `$CATALINA_HOME\conf\catalina.properties` and open `catalina.properties`.
For example, in a typical Tomcat 7.0 installation, the `$CATALINA_HOME` variable may resolve to a path like `C:\Program Files\Apache Software Foundation\Tomcat 7.0`
 - b. Find the line `common.loader=`
 - c. To use a common `dfc.properties` file D2 Config, append the location of `dfc.properties`
For example, `common.loader=<existing paths>,<install path of Documentum>\Config`
 - d. Append the location of the D2 Config configuration files.
For example, `common.loader=<existing paths>,<install path of Documentum>\Config,<install path of Tomcat>\webapps\D2-Config\WEB-INF\classes, <install path of Tomcat>\webapps\D2\WEB-INF\classes`
3. Add or increase the following Java options in your application server environment to instruct the JVM to create permanent generation:

-XX:PermSize=YYYm (for example 256m): sets the initial size of the permanent generation memory space upon startup of Tomcat.

-XX:MaxPermSize=YYYm (for example 256m): sets the maximum amount of permanent generation memory space that can be allocated.

Set **PermSize** to the same value as **MaxPermSize** to allocate the maximum amount of permanent generation memory from startup to help reduce the occurrence of full garbage collection.

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

- **CMSClassUnloadingEnabled: -XX:+CMSClassUnloadingEnabled**

Tuning JVM Garbage Collection for Production Deployments (http://docs.oracle.com/cd/E13209_01/wlcp/wlss30/configwlss/jvmgc.html) contains further information about JVM garbage collection settings.

4. Return to the instructions: [Instructions for Installing D2](#).

Installing D2 on Apache Tomcat for a Linux Environment

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

To update the references:

- a. Navigate to **<install path of Tomcat>/work/Catalina/conf/** and open **catalina.properties**
 - b. Find the line **common.loader=**
 - c. To use a common **dfc.properties** file D2 Config, append the location of **dfc.properties**
For example, **common.loader=<existing paths>,<install path of Documentum>/Config**
 - d. Append the location of the D2 Config configuration files.
For example, **common.loader=<existing paths>,<install path of Documentum>/Config,<install path of Tomcat>/webapps/D2-Config/WEB-INF/classes, <install path of Tomcat>/webapps/D2/WEB-INF/classes**
3. Add or increase the following Java options in your application server environment to instruct the JVM to create permanent generation:

-XX:PermSize=YYYm (for example 256m): sets the initial size of the permanent generation memory space upon startup of Tomcat.

-XX:MaxPermSize=YYYm (for example 256m): sets the maximum amount of permanent generation memory space that can be allocated.

Set **PermSize** to the same value as **MaxPermSize** to allocate the maximum amount of permanent generation memory from startup to help reduce the occurrence of full garbage collection.

You can also configure the clearing of classes:

- CMSPermGenSweepingEnabled: **-XX:+CMSPermGenSweepingEnabled**
- CMSClassUnloadingEnabled: **-XX:+CMSClassUnloadingEnabled**

Tuning JVM Garbage Collection for Production Deployments (http://docs.oracle.com/cd/E13209_01/wlcp/wlss30/configwlss/jvmgc.html) contains further information about JVM garbage collection settings.

4. Return to the instructions: [Instructions for Installing D2](#).

Installing D2 on IBM WebSphere

1. If you are upgrading D2, do the following. Otherwise, go to step 2.
 - a. Use the IBM Administration console to stop the **D2-Config** web application.
 - b. Select **D2-Config** and click **Update**.
 - c. Clear the WebSphere temp folder **<Profile_root>/wstemp**
Where **<Profile_root>** is the directory structure for IBM WebSphere.
 - d. Clear the temp folder **<Profile_root>/temp**
Where **<Profile_root>** is the directory structure for IBM WebSphere.
 - e. Delete the **<install path of Websphere>/webapps/D2** and **<install path of Websphere>/webapps/D2-Config** folders.
2. Connect to the WebSphere Administration console with administrator privileges.
3. Install D2 Config:
 - a. Click **Applications > Install New Application**.
 - b. Click **Browse**, then locate and select **D2-Config.war**.
 - c. Type **/D2-Config** as the context root.
 - d. Click **Next**.
 - e. Change the **Application Name** from **D2-Config_war** to **D2-Config**.
 - f. When you are installing on a cluster:
 - a. Click **Next** to show the **Map modules to server** page.
 - b. Select the **D2-Config.war** module.
 - c. In the **Clusters and Servers** list, hold **CTRL** and select the cluster and web server where you want to deploy the modules that comprise the enterprise application.
 - d. Click **Apply**, then click **Next** to show the **Map virtual hosts for Web modules** page.
 - e. Select the **D2-Config.war** module and click **Next** to show the **Summary** page.
 - f. Review the installation summary. You can click **Previous** to modify any selections.
 - g. Click **Finish** to start the deployment of D2, then click **Save**.
 - g. Follow the wizard, then click **Save** to Master Configuration.
 - h. Navigate to **Applications/Enterprise Applications/<D2-Config>/Manage Modules/<D2-Config>** and set every D2 module to **Classes loaded with local Class**

Loader First (Parent Last) mode. The default is the **Classes loaded with local Class Loader First(Parent Last) mode.**

4. Install D2 Client:
 - a. In the WebSphere Administration console, navigate to **Applications > Install New Application.**
 - b. Click **Browse**, then locate and select **D2.war.**
 - c. Type **/D2** as the context root.
 - d. Follow the wizard until you see **Step 1: Provide options to perform the installation**, then change the **Application Name** from **D2_war** to **D2.**
 - e. Follow the wizard, then click **Save to Master Configuration.**
 - f. Click **Save.**
 - g. Navigate to **Applications/Enterprise Applications/<D2>/Manage Modules/<D2>** and set every D2 module to **Classes loaded with local Class Loader First(Parent Last) mode.** The default is the **Classes loaded with local Class Loader First(Parent Last) mode.**
5. Ensure the **CLASSPATH** used to start the web application server does not reference DFC libraries, because there may be conflicts with the DFC included in the web application.
6. When the configuration files are kept out of the WAR files during extraction, copy the configuration files to the **/WEB-INF/classes** folder or configure the references using the shared environment definition.
7. When the web application server and Document Content Server are on different machines, navigate to **<install path to Websphere>\AppServer\profiles\AppSrv<version>\installedApps\<Cell>\D2.ear\D2.war\WEB-INF\classes** and update the **dfc.properties** file.

Note: If you are using IBM Websphere in a clustered environment, the **dfc.keystore** will need to be copied manually from the Documentum Server machine to the load balancer machine before client log in will be successful.

Copy the **dfc.keystore** from the following location on the Documentum Server machine:
C:\Documentum\config

To the following load balancer machine locations:

D2: C:\Program Files\IBM\WebSphere\AppServer\profiles\Custom01
\installedApps\depmanagerCell101\D2.ear\D2.war\WEB-INF\classes

D2-Config: C:\Program Files\IBM\WebSphere\AppServer\profiles\Custom01
\installedApps\depmanagerCell101\D2-Config.ear\D2-Config.war\WEB-INF
\classes

8. As an optional step, you might want to update the session timeout from its default value of 30 minutes. Note that this must be done in the **web.xml** file in the following two locations:

```
WAS_profile/installedApps/<cellname>/<application.ear>/<webmodule.war>/WEB-INF
WAS_profile/config/cells/<cellname>/applications/<application.ear>/deployments/
<application>/<webmodule.war>/WEB-INF
```
9. Return to the instructions:[Instructions for Installing D2.](#)

Installing D2 on Oracle WebLogic

1. If you are upgrading D2, do the following. Otherwise, skip to step 2.
 - a. Use the Administration console to stop the **D2-Config** web application.
 - b. Select and uninstall **D2-Config**.
 - c. Clear the cache folder `<domain of Oracle WebLogic>/servers/<server name>/cache`
 - d. Clear the temp folder `<domain of Oracle WebLogic>/servers/<server name>/tmp`
2. Perform the following steps:
 - a. Extract **D2-Config.war** to **D2-Config** folder.

Note: Make sure **D2-Config** folder has the same root folder as **D2** folder.
 - b. Open **D2-Config/WEB-INF/weblogic.xml** file for editing and comment **prefer-web-inf-classes** element within the **<container-descriptor>** element as shown:


```
<!--
<prefer-web-inf-classes>true</prefer-web-inf-classes>
-->
```

If the WebLogic version is 10.3.6.0, add the `org.apache.commons` package under the **<prefer-application-packages>** element in addition to the `org.slf4j` package as follows:

```
<prefer-application-packages>
  <package-name>org.slf4j</package-name>
  <package-name>org.apache.commons</package-name>
</prefer-application-packages>
```
 - c. Similarly, extract **D2.war** to **D2** folder and add the above mentioned lines to the **D2/WEB-INF/weblogic.xml** file as well.
 - d. If the WebLogic version is 12.2.1:

In **weblogic.xml**, include `<package-name> javax.ws.rs.core.Application </package-name>` under the **<prefer-application-packages>** element:

```
<prefer-application-packages>
  <package-name>org.slf4j</package-name>
  <package-name>javax.ws.rs.core.Application</package-name>
</prefer-application-packages>
```

In **web.xml**, include the following before DispatchDownload Servlet:

```
<servlet>
  <servlet-name>javax.ws.rs.core.Application</servlet-name>
</servlet>
<servlet-mapping>
  <servlet-name>javax.ws.rs.core.Application</servlet-name>
  <url-pattern>/resources/*</url-pattern>
</servlet-mapping>
```
 - e. Similarly, extract **D2.war** to **D2** folder and add the above mentioned lines to the **D2/WEB-INF/weblogic.xml** file as well.
3. Connect to the WebLogic console with administrator privileges.
4. Install D2 Config:
 - a. Click **Lock & Edit** to open the **Deployments** menu.

- b. Click **Install > Browse**, then select **D2-Config.war** or the extracted folder if Oracle WebLogic is running on Red Hat Enterprise Linux.
 - c. Click **Next**.
 - d. Select **Install the deployment as an application**.
 - e. Follow the wizard, then click **Finish**.
5. Install D2 Client:
 - a. Click **Lock & Edit** to open the **Deployments** menu.
 - b. Extract **D2.war** to **D2** folder. Click **Install**, click **Browse**, then select **D2** folder, and click **Active Change**.

Note: Make sure **D2** folder has the same root folder as **D2-Config** folder.
 - c. Click **Next**.
 - d. Select **Install the deployment as an application**.
 - e. If you are installing D2 on a cluster, select the cluster from the created list of servers.
 - f. Follow the wizard, then click **Finish**.
6. If using Oracle WebLogic 10.3.5, ensure that the main CLASSPATH used to start WebLogic does not contain references to DFC libraries in the **D2-Config** domain of **startWeblogic.cmd**.
7. If the configuration files were kept out of the WAR files during extraction, copy the configuration files to the **D2-Config/WEB-INF/classes** folder or configure the references using the classpath definition.
8. Copy the **D2-Config/WEB-INF/classes/plugin-ins** folder to a location outside of the web application.
9. Update the **D2-Config/WEB-INF/classes/D2-Config.properties** file to reference the new absolute location of installed plug-ins.
10. Return to the [Instructions for Installing D2](#).

Installing D2 on Redhat JBOSS

Note: If you are using JBoss or Wildfly to deploy D2 you will have to update the java policy or permissions.xml first, depending on your version. If you are using JBoss 6.x, update the JRE by editing or creating the java.policy (/lib/security/java.policy/java.policy) file with the following permission:

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

If you are using JBoss 7.x or Wildfly 9.x, add the permissions.xml in ServerApps.ear\META-INF and BPM.ear\META-INF on the Documentum Server to give propagate action for RolePermission:

```
<?xml version="1.0" encoding="UTF-8"?>
<permissions xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/
permissions_7.xsd" version="7">
```

```

    <permission>
      <class-name>
com.documentum.fc.client.impl.bof.security.RolePermission</class-name>
      <name>*</name>
      <actions>propagate</actions>
    </permission>
</permissions>

```

Note: The D2 web app needs to be registered as a privileged client.

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

```

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

```

- a. Make sure `D2.war/WEB-INF/` directory has `jboss-deployment-structure.xml` file.
 - b. Modify the `/root/jboss-eap-6.x/standalone/configuration/standalone.xml` file with the following content:

```

<interface name="public">
<inet-address value="$ {jboss.bind.address:0.0.0.0}" />
</interface>

```
3. Restart the JBoss service.
 4. Update the configuration files inside the `.war` file if necessary.
 5. In the JBoss Administration Console, click **Deployments > Add**.
 6. Select **Upload a new deployment**.
 7. Choose the `D2-Config.war` file.
 8. Click **Finish**.
 9. Return to step 2 and upload the `D2.war` file.
 10. Return to the [Instructions for Installing D2](#).

Installing D2 on VMware vFabric tcServer for Microsoft Windows

1. Copy `D2-Config.war` and `D2.war` to the `<install path of the web application server>\webapps` folder.

2. If the configuration files were kept out of the WAR files during extraction, copy the configuration files to the `<install path to Tc-server>/webapps/<web application>/WEB-INF/classes` folders or configure the references using the classpath definition.
3. Configure the permgen size for the Java Virtual Machine wrapper:
 - a. Navigate to and open for editing the `/conf/wrapper.conf` file.
 - b. Add the following lines:

```
wrapper.java.additional.8="-Xmx512M"
wrapper.java.additional.9="-Xss256K"
wrapper.java.additional.10="-XX:MaxPermSize=1024m"
```
4. Return to the instructions: [Instructions for Installing D2](#).

Installing D2 REST Extension Framework

1. Download the D2 REST .war file from the D2 download site.
2. Copy the .war file to the webserver and extract it to a folder using the command `jar -xvf filename.war`. Extracted contents are the `META-INF`, `public`, and `WEB-INF` folders.
3. Configure `\\WEB-INF\classes\dfc.properties`, which contains information about the repository, machine, username and password.
 - If you are not installing REST on the same server as the Documentum Server, then configure `\\WEB-INF\classes\dfc.properties` to CS.
 - If you are installing REST on the same server as the Documentum Server, then add the following to `\\WEB-INF\classes\dfc.properties`:

```
#include
C:\documentum\config\dfc.properties.
```
4. In order to enable the D2 REST server to load various caches when it starts up, open its corresponding `D2FS.properties` file and update the `LoadOnStartup` parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example: `LoadOnStartup=repo1,repo2`. Note that the corresponding username and password properties for each listed repository need to be set in the D2 global registry keystore. For example,

```
LoadOnStartup.repo1.username=dadmin1
LoadOnStartup.repo1.password=password1
LoadOnStartup.repo2.username=dadmin2
LoadOnStartup.repo2.password=password2
```

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

```
LoadOnStartup.*.username=dadmin
LoadOnStartup.*.password=password
```
5. Navigate to `WEB-INF/classes/rest-api-runtime.properties.template` and note the settings for enabling cross-origin resource sharing and CSRF token settings. Update the same settings to match in the `WEB-INF/classes/rest-api-runtime.properties` file.
6. Verify the D2 REST installation by accessing the URL in this format: `http://ServerIPAddress:PortNumber/ExtractedFoldername/services`. Ensure the D2 REST URL is publicly accessible through one of the deployment options.

Configuring D2 REST Extension Framework

Deploy the REST war on the Application Server and perform the following post installation steps:

1. Update `dfc.properties` in `\WEB-INF\classes`.
2. Update ACS or BOCS in `\WEB-INF\classes\D2FS.properties` for ACS and BOCS environments.
3. Set D2 URL in `WEB-INF\classes\settings.properties`
4. Copy C2 jars (C2-API.jar and C2-Plugin.jar) to `\WEB-INF\lib` to configure the C2 plug-in.

Deploying D2 in a Docker Environment

Preparing the D2 Docker Environment

1. Install the Docker Engine and Docker Compose on a Docker host. Refer to the *Install Docker Engine* and *Install Docker Compose* guides on the Docker support site for more information.
2. Decide whether to install D2 Corepack or Pluspack and whether you want the D2 Documentum Server to run in the Ubuntu or Centos operating system. Then, go to <https://support.opentext.com/>, and download the D2 docker image tars. Here is the list of available tars:

Corepack:

```
D2cs_Corepack_Docker_Centos.tar
D2cs_Corepack_Docker_Ubuntu.tar
D2config_Corepack_Docker_Ubuntu.tar
D2client_Corepack_Docker_Ubuntu.tar
```

Pluspack:

```
D2cs_Pluspack_Docker_Centos.tar
D2cs_Pluspack_Docker_Ubuntu.tar
D2config_Pluspack_Docker_Ubuntu.tar
D2client_Pluspack_Docker_Ubuntu.tar
```

3. Unzip each downloaded tar, which contains the D2 docker image.
4. Unzip the **Scripts.tar** inside **d2cs** tars.
5. Copy the unzipped directories onto the docker host.
6. Load the D2 docker images from tar file into the docker image repository on the docker host. Run the following command for each D2 docker image tar.

```
docker load -i <tar file>
```

Running the D2 Docker Containers

Follow the instructions in the `README.txt` inside the `startAll` folder from the `Scripts.tar` to start all D2 containers.

Add Webapp as DFC Privileged Client

Before you can log into d2 webapp, you need to add the webapp as a DFC privileged client.

1. Get a shell from the webapp container using `docker exec -ti <webapp container name> bash`

2. Navigate to `/usr/local/tomcat/CustomConf` directory. You should see a **dfc.keystore** file in that directory.

3. Run the following command:

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

The resulting output contains detailed information about the certificate stored in the **dfc.keystore** file. Get the value of the CN parameter in the **Owner** field. For example, if the resulting output contains the following line:

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

Then the DFC Client ID is: `dfc_t2zN2bJv8DDyMlZbXD9qJJTWiXA`

4. Go to DA. On the left panel, click **Client Rights Management > Privileged Clients**, then click **Manage Clients** on the top right of the right panel. Find the D2 webapp DFC client id in the list and add it to the right panel, then click **OK**.
5. On the **Privileged Clients** page, right click the newly added DFC client row, then click **Approve Privilege**.

Configuring D2

Installation Validation

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

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

D2 API Libraries in the D2 Installation on the Documentum Server Host

Verify that the host running Documentum Server contains the D2 installation path. For example, by default on a Microsoft Windows operating system, D2 is installed to **C:/Program Files/EMC/D2**

The **D2** folder contains the **config** folder, the **dar** folder, and the following files output by the installer:

- activation.jar
- avalon-framework-<version>.jar
- batik-all-<version>.jar
- bcmail-jdk<version>.jar
- bcprov-jdk<version>.jar
- C6-Common.jar
- commons-compress-<version>.jar
- commons-collections-<version>.jar
- commons-httpclient-<version>.jar
- commons-io-<version>.jar
- commons-lang-<version>.jar
- D2-API.jar
- D2-Constants.jar
- D2.jar
- D2FS-Generated.jar
- D2FS4DCTM-API.jar
- D2-RPS-Connector-API.jar
- D2-Specifications-API.jar

- diff-<version>.jar
- dom4j-<version>.jar
- dtdparser<version>.jar
- ehcache-core-<version>.jar
- fop-hyph.jar
- fop.jar
- geronimo-stax-api_<version>_spec-<version>.jar
- iText.jar
- iTextAsian.jar
- iTextAsianCmaps.jar
- janino-<version>.jar
- jcl-over-slf4j-<version>.jar
- jul-to-slf4j-<version>.jar
- logback-classic-<version>.jar
- logback-core-<version>.jar
- logback.xml
- logback_cs_full.xml
- mail.jar
- ostermillerutils_<version>_for_java_<version>.jar
- PDF-API.jar
- poi-<version>.jar
- poi-ooxml-<version>.jar
- poi-ooxml-schemas-<version>.jar
- poi-scratchpad-<version>.jar
- README.txt
- serializer-<version>.jar
- slf4j-api-<version>.jar
- wfde.jar
- xalan-<version>.jar
- xmlbeans-<version>.jar
- xmlgraphics-commons-<version>.jar

Verify the following Documentum Foundation Services (DFS) files:

- emc-collaboration-services.jar
- emc-collaboration-services-remote.jar

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

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

- certjFIPS.jar
- jsafeFIPS.jar

If you installed FIPS DFS 7:

- certj.jar
- cryptoFIPS.jar

If you installed FIPS DFS 7.1:

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

D2 Libraries in the Java Method Server on the Documentum Server Host

Verify the following files in the `<installation path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\ServerApps.ear\lib\` folder. If you are using Documentum Server 7.1, the `\deploy\` folder is named `\deployments\`

- activation.jar
- avalon-framework-<version>.jar
- batik-all-<version>.jar
- bcmath-jdk<version>.jar
- bcprov-jdk<version>.jar
- C6-Common.jar
- commons-collections-<version>.jar
- commons-io-<version>.jar
- commons-lang-<version>.jar
- D2-API.jar
- D2-Constants.jar
- D2-Widget-API.jar
- D2FS-Generated.jar
- D2FS4DCTM-API.jar
- diff-<version>.jar
- dom4j-<version>.jar
- dtddparser<version>.jar
- ehcache-core-<version>.jar
- fop-hyph.jar
- fop.jar
- geronimo-stax-api_<version>_spec-<version>.jar
- iText.jar
- iTextAsian.jar
- iTextAsianCmaps.jar
- janino-<version>.jar
- jcl-over-slf4j-<version>.jar
- jul-to-slf4j-<version>.jar
- logback-classic-<version>.jar
- logback-core-<version>.jar
- mail.jar

- ostermillerutils_<version>_for_java_<version>.jar
- PDF-API.jar
- poi-<version>.jar
- poi-ooxml-<version>.jar
- poi-ooxml-schemas-<version>.jar
- poi-scratchpad-<version>.jar
- README.txt
- serializer-<version>.jar
- slf4j-api-<version>.jar
- wfde.jar
- xalan-<version>.jar
- xmlbeans-<version>.jar
- xmlgraphics-commons-<version>.jar

Verify the following Documentum Foundation Services (DFS) files:

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

- aspectjrt.jar
- log4j.jar

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

- certjFIPS.jar
- jsafeFIPS.jar

If you installed FIPS DFS 7:

- certj.jar
- cryptoFIPS.jar

If you installed FIPS DFS 7.1:

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

Business Process Management on the Documentum Server Host

If you installed D2 with Business Process Management, verify the following files in the `<installation path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\bpm.ear\lib\` folder. If you are using Documentum Server 7.1, the `\deploy\` folder is named `\deployments\`

- activation.jar
- avalon-framework-<version>.jar
- batik-all-<version>.jar
- bcmath-jdk<version>.jar
- bcprov-jdk<version>.jar
- C6-Common.jar
- commons-collections-<version>.jar
- commons-io-<version>.jar
- commons-lang-<version>.jar
- D2-API.jar
- D2FSDCTM-API.jar
- D2FS-Generated.jar
- diff.jar
- dtdparser<version>.jar
- ehcache-core-<version>.jar
- fop-hyph.jar
- fop.jar

- iText.jar
- iTextAsian.jar
- iTextAsianCmaps.jar
- janino-<version>.jar
- jcl-over-slf4j-<version>.jar
- jul-to-slf4j-<version>.jar
- jcifs-krb<version>.jar
- krbutil.jar
- logback-classic-<version>.jar
- logback-core-<version>.jar
- mail.jar
- ostermillerutils_<version>_for_java_<version>.jar
- PDF-API.jar
- poi<version>.jar
- poi-ooxml-<version>.jar
- poi-ooxml-schemas-<version>.jar
- poi-scratchpad-<version>.jar
- questFixForJDK7
- README.txt
- serializer-<version>.jar
- slf4j-api-<version>.jar
- vsj-license.jar
- vsj-standard <version>.jar
- wfde.jar
- xmlgraphics-commons-<version>.jar

Verify the following Documentum Foundation Services (DFS) files:

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

- jaxb-api.jar
- jaxb-impl.jar
- jaxb-xjc.jar
- jaxws-api.jar
- jaxws-rt.jar
- jsr<version>_api.jar
- jsr<version>-api.jar
- log4j.jar

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

- certjFIPS.jar
- jsafeFIPS.jar

If you installed FIPS DFS 7:

- certj.jar
- cryptoFIPS.jar

If you installed FIPS DFS 7.1:

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

D2 WAR Files on the Web Application Server Host

Navigate to <APPSERVER_INSTALLATION_PATH> and verify that it contains **D2.war** and **D2-Config.war**. For example, in Tomcat the D2.war should be located in the webapps folder.

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

The following information does not apply when you are running a JBoss web application server.

Verify that <APPSERVER_INSTALLATION_PATH>/D2/WEB-INF/lib/ and <APPSERVER_INSTALLATION_PATH>/D2-Config/WEB-INF/lib/ contain the log4j.jar file. D2 Config and D2 Client will fail to connect to the Docbroker repository if the above folders do not exist and you are not using a JBoss web application server.

Configuring D2 Config

1. Navigate to the location of your D2 Config configuration files.
The default location is <install path to web application server>/webapps/D2-Config/WEB-INF/classes
2. Configure **dfc.properties**:

- a. If you want to use a shared set of configurations, configure **dfc.properties** to refer to an existing **dfc.properties**
 By default, **dfc.properties** contains a reference to the Documentum **dfc.properties** file. All settings found in the referenced **dfc.properties** apply to D2 Config. Do not remove the # as the full command is **#include**, and the line is not being commented out.
#include <install path to Documentum>/config/dfc.properties
 - b. If you want to create application-specific settings that override the shared **dfc.properties**, append the settings to the **dfc.properties** found in **<install path to web application server>/webapps/D2-Config/WEB-INF/classes**.
 See **<install path to Documentum>/config/dfcfull.properties** for possible settings.
 - c. Ensure that the **dfc.properties** file being used or referred to addresses the correct docbroker and port:
dfc.docbroker.host=<IP address of the Fully Qualified Domain Name of the docbroker host>
dfc.docbroker.port=<port>
 - d. If your Documentum Server installation uses non-anonymous certificates, copy **dfc.keystore** from the **\$dm_home\dba\secure** folder on the Documentum Server machine. The *OpenText Documentum Server Installation Guide* contains more information on the Documentum Server keystore.
3. Configure **D2-Config.properties** as described in the following table:

Parameter	Description
default_language	Type the two-letter language code to set the default language and prevent users from changing their language option.
forceServerInDocbaseName	Set to true to force connections to use the <repository>@<server> address structure. Note: When forceServerInDocbaseName is set to true , a . (dot) character is allowed in Validator.HTTPCookieValue of the ESAPI.properties .
hideDomain	Set to true to hide the domain on the login dialog box. You can also specify the repository by using the parameter hideDomain.<repository name> .
docbaseFilter	Type a list of repositories, separated by commas, to be hidden from the Repository list box when an end user logs in to D2 Client.

Parameter	Description
temporaryMaxFiles	Type the maximum number of files temporarily stored by D2. Once the maximum is reached, D2 deletes the oldest files.
logLevel	Append one of the following values: <ul style="list-style-type: none"> • all • info • trace • debug • warn • error
logSaveMethod	Set to true to save all event logs from D2Methods in the Temp cabinet of the repository. Configuring logback.xml for the Content Server contains more information. By default this setting is set to false .
D2-BOCS	Set to true to enable BOCS in D2 Client if D2-BOCS is deployed on one or more BOCS servers.
includeAcsServer	Set to true to enable BOCS if D2-BOCS is deployed on the Accelerated Content Services server on the Documentum Server.
proxyClientIpHeader	Set to true to put the client IP address in the header instead of the proxy IP. Use this setting when you have a proxy in your architecture, as by default the proxy replaces the client IP address with the proxy IP. For example, if disabled, you may not be able to select the correct instance of BOCS.

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

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

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

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

5. Set up when and how D2 Config logging events occur by configuring the following elements in **logback.xml**:

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

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

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

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

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

6. In D2 Config, navigate to **Menu > Tools > Reload D2 options** to refresh the options.
7. Return to the instructions: [Instructions for Installing D2](#).

Configuring D2 Client

1. Navigate to the location of your D2 Client configuration files.

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

2. Configure **dfc.properties**:

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

By default, **dfc.properties** contains a reference to the Documentum **dfc.properties** file. All settings found in the referenced **dfc.properties** apply to D2 Client. Do not remove the **#** as the full command is **#include**, and the line is not being commented out.

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

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

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

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

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

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

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

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

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

3. Configure **settings.properties** as described in the following table:

Parameter	Description
transfer.http.compression	Set to true to enable HTTP compression.
login.domain.hide	Set to true to hide the domain for login. You can also specify the repository by using the parameter hideDomain.<repository name> .
error.uncaught.display	Set to true to show uncaught error messages.
uid.session.cookie.timeout	Set the time in seconds that the session remains valid after a user closes or refreshes a browser tab or navigates away from D2 in a browser tab. If the browser itself (not open tabs or windows) is closed the session is lost immediately.
language.user.forced	Append the two-letter language code if you want to force users to access D2 in a specific language and disable language options.
connection.remote.url	Uncomment and type the address of the proxy server to enable content transfer in a reverse proxy setup. http<s>://<proxy or server address>:<port>/D2
browser.folder.limit	Type the limit for the number of folders displayed in any single level of the Repository browser widget. If the end user views a folder containing more items than the limit set, D2 Client shows a More button.

Parameter	Description
login.networklocation.hide	<p>Set to true to show the network location selector.</p> <p>When administrator changes the network location settings in Documentum Administrator, the ACS settings cache is updated every two minutes by default. For testing convenience and to reflect the change immediately in the D2 login dialog, you can add the following line to dfc.properties:</p> <p>dfc.acs.gr.refresh_interval = 0For production machine, we recommend to use the default refresh interval, that is two minutes.</p>
browser.plugin.activex	<p>Set to true to enable users to export content from D2 to desktop using drag and drop feature.</p> <p>This setting enables dragging content to the desktop using an ActiveX plugin. This feature is only supported in Internet Explorer 9 and later.</p>
login.repository.default	<p>Type a repository name to set the default login repository.</p>
login.repository.filter	<p>Type the name of the repositories (separated by comma) that you want to hide from the login screen. For example, Global Registry repository.</p> <p>Note: If you add the default repository name in the list of hidden repositories, the default repository will be hidden from the login screen. That means, login.repository.filter parameter overrides the value of the login.repository.default parameter.</p>
import.prefer.sameproperties	<p>Set to true if user is importing multiple email files with same profile and properties checkbox selected but needs email files to retain their individual attributes (For example, through O2 plugin).</p> <p>Note: File attributes take precedence over inheritance and default value template.</p>

Parameter	Description
browser.plugin.mode	<p>Set to one or more of the content transfer modes. See <i>Documentum Content Transfer Framework Supported Feature List</i> in the Administration Guide for more information.</p> <p>Note: If you do not set a mode, the system defaults to java</p> <ul style="list-style-type: none"> • java • thin • ctf <p>The indicated modes in settings.properties determines which clients the users will be able to choose from in the D2 Client > User Settings. In addition, the order will determine the fallback process D2 will follow. For example, browser.plugin.mode=ctf, java, thin would attempt ctf first, then fall back to thin, while java, ctf, thin would attempt java first, then fall back to thin. See the settings.properties file for further details.</p>
download.folderexport.batchsize	<p>Set the maximum number of document download URLs returned in the initial folder export request. This limit prevents the client from timing out the request when processing a very large number of documents. Default is 500 documents.</p>
checkin.rlockmachine.check	<p>Set rlockmachine validation when checking in files. Introduced for Mapped Network Drives. If true, rlockmachine validates when checking in files. If false, no rlockmachine.</p> <p>The check will be performed when checking in files using the java plugin. If the line is commented out, the effective value is true.</p>

The *Drag and Drop Behaviors* section of the *OpenText Documentum D2 User Guide* provides details about the behavioral changes in D2 for drag and drop.

4. Configure client-side compression in **settings.properties** as described in the following table:

Parameter	Description
applet.download.compression.enable	Set to true to enable compression for downloads.
applet.upload.compression.enabled	Set to true to enable compression for uploads.
applet.upload.compression.threshold	Set a value at which the compression begins. For example, applet.upload.compression.threshold = 1024
applet.upload.compression.extensions	Type a list of file format extensions separated by commas that undergo compression. For example, applet.upload.compression.extensions = doc,docx,xls,xlsx,ppt,pptx,pdf,txt

5. Configure **D2FS.properties** as described in the following table:

Parameter	Description
cacheDocumentumDql	Type a DQL query to find content to simulate location computation. For example: cacheDocumentDql = dm_document where r_content_size > 102400 order by r_content_size asc
cacheLocations	Type a list of cache locations separated by a comma. By default, this parameter uses the local IP address. For example: cacheLocations = network1,network2
compressedExtensions	Type a list of file extensions separated by a comma. For example: compressedExtensions =doc,docx,xls,xlsx,ppt,pptx,pdf,txt
hideDomain	Set to true to hide the domain for login. You can also specify the repository by using the parameter hideDomain.<repository name> .

Parameter	Description
maxResultSetSize	<p>Type a value to limit the result set of all queries used in populating the Users and Groups widgets as well as user and group selection lists in property dialog boxes. Use this parameter to avoid performance problems associated with large result sets.</p> <p>The default value is 1000.</p> <p>Note: This is applicable to similar widgets such as, repository browser widget.</p>
contentTransferUrlTicketTimeout	<p>Type a value to set the time in minutes that a D2 download URL remains valid.</p>
pluginsOrder	<p>Type a list of plug-ins by name to force the order in which they are loaded.</p> <p>For example, if you have plugin1 that computes data during a property save and plugin2 that verifies data during a property save, you want verification to occur after computation. In this example, set the line as pluginsOrder=plugin1,plugin2</p>
maxTempFiles	<p>Set the maximum number of temporary files that D2 can store in the temporary location of the application server. The default value is 10240.</p>
cleanUpFraction	<p>Set the fraction value that is used to calculate the cleanup size, after the number of temporary files in the temp location exceeds the maximum number of files.</p> <p>For example, if the cleanUpFraction is set to 0.1 then the cleanup size is calculated as maxTempFiles*0.1. The value of cleanUpFraction should be between 0.0 and 1.0, excluding 0 and 1. The default value of cleanUpFraction is 0.1.</p> <p>Note: It is advised to set this value carefully after considering the value of maxTempFiles. Setting this value high may cause a delay in response when the system is performing cleanup.</p>

Parameter	Description
localFallback	<ul style="list-style-type: none"> • If localeFallback language is specified and browser language is not installed on Documentum Server listed in DD_LOCALES, D2 falls back to the specified language. • If not specified, D2 uses the D2-Config labels for the user locale regardless of DD_LOCALES and returns the config ID when the localized configuration is missing in the existing behavior. • If localeFallback language is specified, setting should contain only one language code that is #localeFallback=en
workflowWithTBO	<p>Set to true, if TBO is attached to the workflows.</p> <p>Note: In case of sequential tasks, update performer operation is disabled for workflows with TBO.</p>
processXploreResultSet	<p>Set this flag to true if post processing of the Xplore result set is needed. Setting this to true is useful if there is a need to fetch checkout status of a document in search result set. Note that this may slightly degrade the performance of D2 search. If unset, the default value is taken as false.</p>
allowRenditionRequest	<p>Control multiple rendition requests for the same document. The default is false, which disallows multiple requests. signoff directs D2 to check if the document is already signed off by the rendition server, then allows the request. If the sign off is not in place, multiple requests are denied. true allows D2 to create a new rendition request for the same document without restriction.</p>
blockViewerRenditionRequest	<p>Controls the queue status of the rendition in dmi_queue_item from the PDF viewer. true blocks the rendition. Default is false.</p>
showD2TasksOnly	<p>Set this flag to true to display tasks triggered from D2. Setting the flag to false displays tasks triggered from products other than D2.</p>

Parameter	Description
simpleSearchDql	<p>Allows you to make search options configurable when Enable full text search is not enabled in D2 Config's Interface > Search menu.</p> <p>To override the default search dql, specify simpleSearchDql=<your_custom_dql> using \$value(search_term) to reference the search term.</p>
objectCreationLocation	<p>Allows you to specify a fixed location for initial object creation.</p> <p>By default, Documentum Server implicitly links new objects to the user's defined home cabinet/folder. If users do not have WRITE access to their home cabinet/folder, this setting can be used to specify a temporary location where objects will be created. When D2 completes the creation process, the object will be re-linked to its final location as usual (for example, through auto-linking or other D2 functionality).</p> <p>Note: This folder must be pre-existing and all users must have at least WRITE permission to the folder. It will not be created by D2.</p>
loadBalancedContentServer	<p>Flag that identifies your Documentum Server environment as being load balanced. Default is false (your environment is not load balanced).</p> <p>Set flag to true if you have employed Documentum Server load balancing.</p> <p>Note: If you have set up a high availability multi-content server environment, ensure that this flag is set to true to avoid [DM_SESSION_E_AUTH_FAIL] errors from appearing when users view PDFs in the PDF Viewer widget.</p>

Parameter	Description
showHomeCabinet	Configures the users Home Cabinet to be displayed in the Repository, DocGallery (Thumbnail) and Document list widgets. The values are true , false and default . The default behavior is the same as false and means existing 4.2 behavior is retained. When set to true it causes the Home Cabinet to appear on DocGallery (Thumbnail) and Doc List widgets and in Repository Widgets where (via configuration) there is no start path defined.
checkForMissingDocInWorkflow	Checks if a document in a workflow is deleted if the value is set to true . If set to false , this check is skipped.
irmProtectedFormats	Sets Information Rights Management for selected file types, including: pdf , msw8 , excel18book , ppt8 , msw12 .
aspectsRequiringAppServerForUpload	Lists the aspect names that require upload through the D2 app server.
doclistFilterPropertyNames	Sets Non-repeating string-valued properties that the end user can use to filter the contents of a folder or cabinet in the Doclist widget. If unset, the default value is object_name,title,subject,a_status,r_creator_name,r_modifier Note that the order in which properties appear in the combo box dropdown is given by the order indicated here.
Validator.format	Validates whether given value can be a dm_format.name value or dm_sysobject.a_content_type value This regular expression requires that the value have at least 1 character and at most 32 characters, where every character being a letter, digit, underscore, or hyphen. dm_format.name values can have up to 64 characters, but dm_sysobject.a_content_type values must have no more than 32 characters. If commented out, no validation will be performed, but this introduces a DQL injection security vulnerability.

Parameter	Description
propertyPageConstraintValidation	Default is false (disabled). Set to true to enable the validation constraint. This setting ensures that disabled, hidden, or read-only (immutable) properties cannot be saved if the initial value sent to the requesting system is not the same value returned.
maxBrowserWidgetResultSetSize	Sets the maximum result set size for the Browser widget when users are browsing into the tree : If unset, the default value is maxResultSetSize .
maxUploadRequestSize	Sets the maximum upload request size in bytes (default value is 16GB) This enables the prevention of denial of service attacks launched by uploading very large files. Note: File upload to the D2 app server is implemented using a multipart form post, so the size of a file upload request body will be slightly larger than the file itself. This setting is used when the multipart form post request is parsed. If the value of the Content-Length request header exceeds the maximum upload request size, the request is rejected. If the value of this setting is zero or negative, no restriction is imposed on the size of a multipart form request.
objectCreationLocation	Specifies a fixed location for initial object creation. By default, the Documentum Server implicitly links new objects to the user's configured home cabinet/folder. If users do not have WRITE access to their configured home cabinet/folder, this setting can be used to specify a temporary location where objects will be created. When D2 completes the creation process, the object will be re-linked to its final location as usual (for example, through auto linking or other functionality) Note: This folder must exist and all users must have at least WRITE permission to the folder. The folder will not be created by D2.
maxRecycleBinWidgetResultSetSize	Sets the maximum result set size for the recycle bin widget before filtering is used. If unset, the default value is maxResultSetSize .

Parameter	Description
cancelCheckoutVdConversionToSimpleDoc	Converts a virtual document to a simple document on the cancelCheckout action using context menu option. If specified and set to true , a virtual document is converted to a simple document on the cancelCheckout action. If not specified or set to false a virtual document is not converted to a simple document on the cancelCheckout action.
cancelCheckoutVdConversionToSimpleDoc	If specified and set to true a user with Version permit is not allowed to convert a simple doc to virtual document and gets a failure message of insufficient access. If not specified or set to false a user having Version permit on a simple document can convert it to a virtual document. On adding a grandchild_doc, the child_Doc is converted to a virtual document.
useSQLServerPerfHint	Replaces "RETURN_TOP" with "SQL_DEF_RESULT_SET" hint in doclist DQL to improve SQLServer performance. Default is false . Set to true to turn on the hint, and add the following setting in server.ini: "SQL_DEF_RESULT_SET_AND_OBJECT_BASED=1"
allowThinClientDirectBocsDownload	Allows direct BOCS download in thin client mode.

6. In order to enable the D2 app server to load various caches when it starts up, open its **D2FS.properties** file and update the **LoadOnStartup** parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example: `LoadOnStartup=repo1,repo2`. Note that the corresponding username and password properties for each listed repository need to be set in the D2 global registry keystore. For example,

```
LoadOnStartup.repo1.username=dadmin1
LoadOnStartup.repo1.password=password1
LoadOnStartup.repo2.username=dadmin2
LoadOnStartup.repo2.password=password2
```

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

```
LoadOnStartup.*.username=dadmin
LoadOnStartup.*.password=password
```

7. Set up when and how D2 Client logging events occur by configuring the following elements in **logback.xml**:

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

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

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

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

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

8. Return to the instructions: [Instructions for Installing D2](#).

Configuring D2 Java Method Server

1. Stop the JMS.
2. Navigate to the **APP-INF/classes** folder of the Documentum Server JMS.
3. Create **D2-JMS.properties** if the file does not exist.
4. To configure the order that D2 applies autolink and security to content, add or set the line:
forceLinkAfterSecurity = <true or false>
Where **true** forces D2 to apply Autolink rules to content before applying Security, and **false** forces D2 to apply Security rules to content before applying Autolink.
5. Restart the JMS.
6. Return to the instructions: [Instructions for Installing D2](#).

File Transfer Modes

You can configure how content transfer is performed by D2 browser clients. D2 offers three options:

- Java applet mode: this mode is available when the user's browser supports java applets. If so, the X3 applet will be used to perform content transfer operations such as download, import, checkin, folder export, folder import, etc. and work with native annotations.
- CTF plugin mode: this mode is available when the user's browser is either IE 11, Chrome, or Firefox on Windows or Safari on Mac OSX. The CTF browser extension and native application will be used to perform content transfer operations such as download, import, checkin, folder export, folder import, and work with native annotations. Note that installation of the CTF plugin on IE 11 requires that the end user have administrative privilege on the local machine.
- Thin client mode: this mode is available for all supported browsers. Note that features such as native annotations, folder import, and folder export are not available because these require either the java applet or CTF plugin.

Note: D2 functions such as the various File Transfer modes require all components to be at the same "bitness" level. Transferring a 32 bit Outlook file requires 32 bit browser and 32 bit Java parity, while 64 bit Outlook requires 64 bit browser and 64 bit Java.

Installing DCM Browser Extension (Internet Explorer)

Make sure the following prerequisites are met on the client machine:

- Install Microsoft .NET 4.5 or later.
 - Make sure you have administrator privileges to install the Browser Helper Object (BHO).
 - Disable the pop-up blocker.
 - Remove the TabProcGrowth entry (if available) from **HKEY_CURRENT_USER** or **HKEY_LOCAL_MACHINE/Software/Microsoft/Internet Explorer/Main/TabProcGrowth**.
 - Disable the **Enable Enhanced Protected Mode** option in **Internet Options > Advanced > Security**.
 - Enable the **Third-Party Browser Extensions** option in **Internet Options > Advanced > Browsing**.
 - Make sure the **Delete Browsing History** option is unchecked in **Internet Options > General**.
 - Enable the **Automatically Detect Intranet Network** option in all zones in **Internet Options > Security > Local intranet > Sites**.
 - (Only for Kerberos SSO environment) Disable the **Display intranet sites in Compatibility View** and **Use Microsoft compatibility lists** options.
1. Launch Internet Explorer and log in to D2.
 2. When prompted, install the content transfer browser extension.
 3. Click **Run** and install the BHO and native application through a single installer.
 4. When the installation has completed, click **Tools > Manage Add-Ons**.
 5. Verify the status of OpenText Documentum Webtop Browser Helper Object. If the status is not enabled, enable it manually.
 6. Restart the browser and log into D2.

Note: Internet Explorer launches a separate CTF instance for each browser tab, so the possibility of multiple CTF applications running simultaneously should be monitored. Users are suggested to leave only one browser tab open at a time.

Installing DCM Browser Extension (Firefox)

Make sure the following prerequisites are met on the client machine:

- Disable the pop-up blocker.
1. Launch Firefox and log in to D2.

2. When prompted to install the content transfer browser extension, click **Install**.
3. When prompted that you are prevented from installing the content transfer browser extension, click **Allow** and then click **Install**.
4. Restart the browser and log into D2.
5. When prompted to install the native client application, click **Save File**, click on the saved file, and click **OK** once the installation is complete.
6. Restart the browser and log into D2.

Installing DCM Browser Extension (Chrome)

Make sure the following prerequisites are met on the client machine:

- Disable the pop-up blocker.
 - Have access to the Chrome Store to install the content transfer browser extension.
1. Launch Chrome and log in to D2.
 2. When prompted to install the content transfer browser extension, click **Install**.
 3. On the OpenText Documentum Client Manager Chrome App Store page, click **ADD TO CHROME** and then click **Add extension**. Close the dialog once you are prompted that the extension has been added.
 4. Restart or refresh the browser and log into D2.
 5. When prompted to install the native client application, click **Save File**, click on the saved file, and click **OK** once the installation is complete.
 6. Restart or refresh the browser and log into D2.

Installing DCM Browser Extension (Safari)

1. Launch Safari and log in to D2.
2. When prompted to install the content transfer browser extension, click **Install**.
3. Restart or refresh the browser and log into D2.
4. D2 automatically downloads the native application installer. Click on the saved DMG file, and move DCM application to Applications folder as indicated.
5. Navigate to the Applications folder. Double-click and launch the DCM application.
6. When prompted to install the native client application, click **Save File**, click on the saved file, and click **OK** once the installation is complete.
7. Restart the browser and log into D2.

Note: Some additional notes for Safari:

- Launching the DCM application from the Applications folder is only necessary on first installation. On subsequent restarts, the DCM application will be automatically launched and running in background.
- If GateKeeper settings are set to **Mac App Store and identified developers** in the end user's Mac, on install of DCM App, the app has to be opened with following steps.
 1. In Finder, Control-click or right click the icon of the app.
 2. Select **Open** from the top of contextual menu.
 3. Click **Open** in the dialog box. If prompted, enter an administrator name and password.

Uninstalling Documentum Client Manager Browser Extensions

Follow these procedures to uninstall Documentum Client Manager (DCM) Browser Extensions manually:

- *Google Chrome:* Navigate to `chrome://extensions/` in the Chrome browser and click the **Remove from Chrome** icon next to the *OpenText Content Transfer Extension*.

Note: Uninstalling the Chrome extension will not uninstall the DCM Native Application.

- *Internet Explorer:* An MSI-based installer is used for installing both the IE extension and the DCM native app. Uninstall through **Control panel > Programs and features > Uninstall a program > OpenText Content Transfer Helper**.

Note: Uninstalling the IE extension will also uninstall the DCM Native Application.

- *Firefox:* Navigate to **about:addons** in the Firefox browser and execute the **Remove** action.
- *Safari:* Navigate to the Safari menu dropdown, then **Preferences > Extensions** to find the extension for removal.

Uninstalling Documentum Client Manager Native Application

No formal uninstaller is provided with the DCM native application, follow these instructions to delete the DCM client footprint from the file system and registry:

1. Delete the registry entry at the following location:
`HKEY_CURRENT_USER\Software\Google\Chrome\NativeMessagingHosts\com.emc.ctf.native.app.<version>`
2. Delete the files/folders from the file system at the following locations:
 CTF Native Application installation location:
`%LOCALAPPDATA%\EMC\ContentXfer\com.emc.ctf.native.app\<version>`
 Native application log location:

%temp%\ContentTransferDaemon.log

Note: For Mac based DCM installation, DCM application can be deleted from **Applications** to cleanly uninstall the native application.

Configuring File Transfer Modes

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

`browser.plugin.mode=<modes>`

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

- `java`
- `thin`
- `ctf`
- `java,thin`
- `ctf,thin`
- `java,ctf`
- `ctf,java`
- `java,ctf,thin`
- `ctf,java,thin`

The behavior for each of these setting values is given in the following table:

<code>browser.plugin.mode</code>	Behavior
<code>thin</code>	Always use thin client mode for all users.
<code>java</code>	Always use java mode for all users. Fallback to thin client mode if the browser supports java mode but cannot load java plugin.
<code>ctf</code>	Always use ctf mode for all users. Fallback to thin client mode if the browser supports ctf mode but cannot load ctf plugin.
<code>java,thin</code>	Default to java mode for all users. Fallback to thin client mode if the browser supports java mode but cannot load java plugin. Users can choose to use java or thin client mode from the user options dialog.
<code>ctf,thin</code>	Default to ctf mode for all users. Fallback to thin client mode if the browser supports ctf mode but cannot load ctf plugin. Users can choose to use ctf or thin client mode from the user options dialog when both modes are supported by browser.

browser.plugin.mode	Behavior
java,ctf	Default to java mode for all users. Fallback to thin client mode if the browser supports java mode but cannot load java plugin. Users can choose to use java or ctf mode from the user options dialog when both modes are supported by browser.
ctf,java	Default to ctf mode for all users. Fallback to thin client mode if the browser supports ctf mode but cannot load ctf plugin. Users can choose to use ctf or java mode from the user options dialog when both modes are supported by browser.
ctf,java,thin	Default to ctf mode for all users. Fallback to thin client mode if the browser supports ctf mode but cannot load ctf plugin. Fallback to thin client mode if the browser supports java mode but cannot load java plugin and user has chosen java mode in user settings. Users can choose to use ctf or java or thin client mode from user options dialog for those modes supported by browser.
java,ctf,thin	Default to java mode for all users. Fallback to thin client mode if the browser supports java mode but cannot load java plugin. Fallback to thin client mode if the browser supports ctf mode but cannot load ctf plugin and user has chosen ctf mode in user settings. Users can choose to use java or ctf or thin client mode from user options dialog for those modes supported by browser.

Browser: OS \ Mode	Thin	Java	CTF	Note
IE 11	Yes	Yes	Yes	
Edge	Yes	No	No	1
Firefox	Yes	Yes	Yes	
Chrome	Yes	No	Yes	1
Safari: Mac_OSX	Yes	Yes	Yes	
Safari: Mac_IOS	Yes	No	No	2

(1) Chrome and Edge do not support java applets, and Edge does not support the CTF plugin. D2 will fall back to thin client mode appropriately when java or ctf has been specified in the value of the browser.plugin.mode setting as described above.

(2) If browser.plugin.mode contains java or ctf, then D2 will silently continue to run in thin client mode. Safari running on Mac_IOS does not support the java or ctf plugin.

Note: There are two fields in the D2 user settings UI pertaining to browser plugin mode:

Current Browser Plugin label - shows which mode is currently being used by the browser.

Update Browser Plugin combobox - choice list is given by mode(s) specified in the browser.plugin.mode setting value, with those modes that are not supported by the current browser omitted. For example, java mode is not supported by Chrome or Edge, and ctf mode is not currently supported by Edge.

This choice is persisted with the browser and not in the repository. It is cleared when cookies are cleared in the browser.

3. If you upgraded the ACS or BOCS server to a version that supports content transfer in thin client mode or using direct URL, open **D2FS.properties** in a text editor and set the following line:

allowThinClientDirectBocsDownload=true

The default parameter value of **false** disables direct downloads using an ACS or BOCS server when D2 is in thin client mode.

4. Return to the instructions: [Instructions for Installing D2](#).

Configuring logback.xml for the Documentum Server

Configure logging for:

- The Documentum Server using the **logback.xml** located in the install path to D2.
- The Java Method Server using the **logback.xml** located in **<install path of JMS>/Dctm-Server_MethodServer/deploy/Server-Apps.ear/**

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

1. If the JMS file is named **logback_jms_full.xml**, rename it to **logback.xml**.
2. To change when and how D2 JMS logging events occur, configure the following elements:

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

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

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

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

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

- Return to the instructions: [Instructions for Installing D2](#).

Removing Debug Logs from the JBoss 7.1.1 JMS Log

- Navigate to `<install path to Documentum Server>/<Java Method Server>/server/DctmServer_MethodServer/configuration/` and open `standalone.xml` in a text editor.
- Under the `<console-handler>`, change the value of `<level name="<log level>" />` to `warn`.

For example:

```
<console-handler name="CONSOLE">
  <level name="warn" />
</console-handler>
```

Configuring Documentum Server server.ini

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

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

- Set the value `mail_notification` to `TRUE` to enable mail notifications for queue work items or events for the Documentum Server.
If the parameter is missing you do not need to add it because the default value is `TRUE`.
- If not present, add or set the line `return_top_results_row_based=false` to the `[SERVER_STARTUP]` section.
This setting prevents repeating attributes from being returned as individual rows in lists such as advanced searches, property pages, and repository browser widgets.
Note: D2 installation configures this setting automatically. Perform the above step to change the behavior post-installation.
- If you are working in a clustered Documentum Server environment, add or set the line `upd_last_chg_time_from_db` to `TRUE` for each running Documentum Server. This setting specifies that all Documentum Servers in a clustered environment have timely access to all changes in group membership.
- Restart the Documentum Server.
- Return to the instructions: [Instructions for Installing D2](#).

Configuring D2 Auditing

If you are installing D2 or configuring audit for the first time, create a registered table to allow queries on the audit trail and the ability to read audit information related to deleted content. If you are upgrading D2, you do not need to perform these steps.

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

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

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

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

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

3. Return to the instructions: [Instructions for Installing D2](#).

Configuring Application Server Pooling Session

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

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

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

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

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

For example, you can set `duration` to 300 for a duration of 5 minutes.

3. Return to the instructions: [Instructions for Installing D2](#).

Configuring Compatibility with Documentum Information Rights Management (IRM)

You can configure the compatibility of D2 with Documentum IRM to provide support for information rights management in D2 Client. Documentum IRM adds security and controls to content in the D2 repository.

1. Install IRM Server on a separate server machine. The *IRM Server (Windows) Installation Guide* contains instructions for installing IRM Server.

2. Install IRM Services for Documentum on the same web application server as D2 Client. The *IRM Services for Documentum Installation and Configuration Guide* contains instructions for installing IRM Services for Documentum.
3. Deploy **IRM Services.dar**. The *IRM Services for Documentum Installation and Configuration Guide* contains instructions for locating and deploying **IRM Services.dar**.

Configuring the D2EventSenderMailMethod

When D2 is installed into a repository, **D2EventSenderMailMethod** updates the **mail_method** attribute of **dm_server_config** to capture and process events related to D2. If the event is not related to D2, D2 uses the **dm_event_sender** dmbasic method.

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

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

Configuring D2 to use CTS for Fast Web-Enabled C2 Renditions

To set up the CTS client environment so that linearized PDFs can be generated in D2, follow the steps below:

1. Create a folder (for example, **realtimeclient_config**) somewhere in the client file system, then set up the following structure within the directory:
 - a. Copy **aek.key** from CTS host (from %CTS%\config) to the folder (this will be used in **preferences.xml** as part of **<AekFilePath>**)
 - b. Copy **mpassword.txt** from CTS (from %CTS%\docbases\<<your_docbase>\config\pfile) to the **pfile** folder. This will be used in **preferences.xml** as **ServerProperty passwordFile**
 - c. Create an empty **cache** folder. This will be used in **preferences.xml** as **ServerProperty Cache**
 - d. Create a **preferences.xml** file. A sample version of this file suitable for editing can be copied from the CTS server at %CTS%\config. Make sure you update the file with relevant **path/docbase** values for the **ServerProperty Key** cache value, **AekFilePath**, **LoginContext DocbaseName**, **ServerProperty Key** administrator value, and

ServerProperty Key password file value (values that must be updated in the file are shown below):

```
<ServerProperty Key="Cache" Description="The Temporary Cache Directory"
Value="C:/realtimeclient_config/cache" />
<AekFilePath>C:/realtimeclient_config/aek.key</AekFilePath>
<LoginContext DocbaseName="Irx64">
<ServerProperty Key="userName" Value="Administrator"/>
<ServerProperty Key="passwordFile" Value="C:/realtimeclient_config/pfile/mspassword.txt"/>
```

Note: In case of multiple repositories configured with CTS, you might need to add multiple login context nodes under the **<Repositories>** node in the `preferences.xml` file. Also you have to create a separate `pfile` folder for `mspassword.txt` per repository. For example, if you have two docbases, `repo1` and `repo2` configured with your CS, you should create relevant folders. Also, copy `mspassword.txt` from the corresponding docbase folder on the CTS machine.

2. Create an environment variable called `CTS_CONFIG_LOC` with the path of the root folder that contains the `preferences.xml` file as the value. The value would be the path of the `realtimeclient_config` folder; `C:\realtimeclient_config`).
3. Check the **Fast Web Compatibility** checkbox in D2 Config's **C2** menu selections **Rendition configuration**, **Print configuration**, **Export configuration**, or **View configuration** to turn on linearized PDF generation. In Advanced Publishing, the checkbox can be found on the **PDF configuration** tab.

Enabling POP3S Mail Configuration

Follow these steps to enable POP3S mail configuration support in D2:

1. In D2-Config under **Tools > Email** ensure **Use SSL** checkbox is checked.
2. Create a file named `pop_c6.properties` with the following text:

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

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

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

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

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

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

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

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


```
<!--<resource-root path="javax.mail-1.5.3.jar"/>--> <resource-root path="mail.jar"
```
7. Make JCE Enabled. Copy US_export_policy.jar and local_policy.jar to **%JAVA_HOME%\jre\lib\security**
8. Import Certificate. Copy DigiCert, DigiCertCloudCert and Office365 Cert from C:\, then Run the following command:

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

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

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

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


Best Practices

Enabling Compression at the Application Server when using Apache Tomcat or JBoss

If you deployed D2 on Apache Tomcat or JBoss, you can enable compression to reduce the amount of data transferred from the server to the clients. This setting improves end-to-end response time, especially under WAN conditions, and lowers the throughput (bytes/seconds) for the same transaction rate.

The compression rules on the application server do not apply for content transfer cases, such as importing or exporting a 300 MB document. D2 uses a built-in compression mechanism for uploading and downloading content.

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

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

Optimizing Performance for Widgets and Large Numbers of Content

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

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

Use Google Chrome Frame with Microsoft Internet Explorer browsers to significantly improve browser-side performance.

1. Navigate to and open `<install path of D2 Client>/WEB-INF/classes/D2FS.properties`
2. To configure the maximum result size for the User, Group, Doclist, Thumbnails, and List assistance widgets, set the following parameter. If the number of content found is larger than the threshold set with this parameter, D2 Client shows a filter field at the top of the widget.

The end user can type keywords in the filter to search for the object if it is not included in the truncated result.

maxResultSetSize=<number of results>

3. To configure a maximum result size specifically for the Doclist widget, set the following parameter:

maxDoclistWidgetResultSetSize=<number of results>

You can limit the number of items shown per page using the iapi:

?,c,update d2c_preferences object set pagination_enabled=1

?,c,update d2c_preferences object set pagination_number=<number of items per page>

4. To configure a maximum result size specifically for the User and Group widgets, set the following parameter:

maxAdminWidgetResultSetSize=<number of results>

5. To configure a maximum result size specifically for the Thumbnails widget, set the following parameter:

maxDocgalleryWidgetResultSetSize=<number of results>

6. To configure a maximum result size specifically for the dialog box shown when loading content, set the following parameter:

maxListAssistanceResultSetSize=<number of results>

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

browser.folder.limit=<number of folders>

Improving Content Transfer Performance

1. Enable compression of content:
 - a. Navigate to and open **<install path of D2 Client>/WEB-INF/classes/settings.properties**
 - b. Set the following parameters:

```
applet.download.compression.enabled=true
applet.upload.compression.enabled=true
applet.upload.compression.threshold=<files larger than this size in bytes are compressed>
applet.upload.compression.extensions=<list extensions applicable for compression separated by a comma>
```
2. Increase the socket buffer size when using Apache Tomcat or JBoss because the default buffer size is usually too small for downloading a large document under WAN conditions:
 - a. Navigate to and open **<TOMCAT_HOME>/conf/server.xml**
 - b. Configure the threshold of the content size and the type of content to be compressed:

```
<Connector port="<server port>" protocol="HTTP/1.1"
        socketBuffer="<buffer size in bytes>"
    "/>
```

3. Configure the Apache Tomcat NIO:

- a. Navigate to and open `<TOMCAT_HOME>/conf/server.xml`
- b. Configure the threshold of the content size and the type of content to be compressed:

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

General Tuning Tips

Tune the following parameters, which are not specific to D2, according to varied workload.

When using an Oracle application server:

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

On the Documentum Server:

- Modify `server.ini` and set the **concurrent_sessions** parameter.

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

On the web application server:

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

Note: Java heap size values:

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

Configuring Authentication

Single Sign On (SSO) Authentication for D2

D2 offers authentication for various forms of SSO, such as NTLM, Kerberos, and TrustedReverseProxy.

Note: The TrustedReverseProxy form of SSO allows integrations with RSA AccessManager, CA SiteMinder, and IBM WebSEAL.

The NTLM authentication uses DFC Principal Authentication mode which creates Documentum Server login tickets on behalf of the authenticated users. This authentication is known as SSO authentication for application server. This authentication does not require any authentication plugins or configuration on the Documentum Server.

The Kerberos and TrustedReverseProxy authentications can also use the DFC Principal Authentication mode as well as a different mode for authenticating users on the Documentum Server. This mode of authentication is known as SSO authentication for both application server and Documentum Server. This authentication requires an appropriate authentication plugin to be installed and configured on the Documentum Server. In this authentication mode, D2 uses an SSO token issued by the SSO provider (Kerberos, RSA Access Manager, or CA SiteMinder) as a credential when obtaining a session on the Documentum Server on behalf of a user, and the Documentum Server authenticates this token before granting a session to the D2 application server.

Note: The IBM WebSEAL SSO provider does not have any authentication plugin for the Documentum Server. You can configure TrustedReverseProxy authentication for IBM WebSEAL using SSO authentication for application server only.

To configure SSO authentication for D2, configure the desired form of SSO (NTLM, Kerberos, and TrustedReverseProxy) in the `shiro.ini` file, add the administrator login name and password for each relevant repository to the D2 keystore. By default, the `shiro.ini` file is configured to allow D2-Config and D2Methods running in the Documentum Java Method Server (JMS) to bypass SSO when making certain requests to the D2 application server. Do not alter these settings.

Configuring Microsoft Windows NT Unified Logon (NTLM)

1. In your Active Directory Server, create a user with the same name as the computer hosting your application server.
2. Use Documentum Administrator or D2 Client to create a user with the same name as in Step 1 in your repository.
3. If you are using Microsoft Windows 7:
 - a. Log in to the client machine with Administrator privileges.
 - b. Run `secpol.msc`.

- c. Navigate to **Security Settings > Local Policies > Security Options > Network Security: LAN Manager authentication level**.
 - d. From the list box, select **Select NTLM response only**.
 - e. Click **OK**.
 - f. Restart the computer to enable the new group policy.
 - g. Log in to the client machine with the user created in Step 1 to access the application.
4. Perform the steps in [Adding Administrator Credentials to the D2 keystore for SSO, page 91](#)
 5. Perform the steps in [Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server](#) to configure the shiro.ini file.
 6. Navigate to `webapps/D2/WEB-INF/classes/` and open `shiro.ini`. If `shiro.ini` does not exist, create a copy of `shiro_base.ini` and rename it as `shiro.ini`. Make the following changes to `shiro.ini`:

```
[main]
X3-NTLM = com.emc.x3.portal.server.filters.authc
.X3NtlmHttpAuthenticationFilter
X3-NTLM.defaultRepository=<default repository>
X3-NTLM.domainController=<domain controller>
X3-NTLM.domainName=<domain name>
X3-NTLM.domainUser=<login name of domain user>
X3-NTLM.domainPassword=<password of domain user>
##Authentication type
/** = X3-NTLM
```

Note: OpenText strongly recommends to use the encrypted password for `X3-NTLM.domainPassword` parameter. Use the `com.emc.d2.api.utils.GetCryptedPassword` `java` command line utility available in `D2-API.jar` to encrypt the password. To encrypt a password, add `D2.jar` to the system `CLASSPATH` and use the command `java.com.emc.d2.api.utils.GetCryptedPassword`.

7. Return to the [Instructions for Installing D2](#).

Configuring Kerberos

1. In your Active Directory Server, create a user with the same name as the computer hosting your application server.
2. Right-click the newly created user and select `Properties` from the context menu.
3. In the `<User> Properties` dialog box, select the `Account` tab.
4. According to your operating system and the encryption algorithm that you require, select one of the following encryption algorithms under `Account options`:
 - a. On a *Non-Windows 2008 machine* select the **Use DES encryption types for this account** checkbox.
 - b. On a *Windows 2008 machine*, select the **This account supports Kerberos AES 128 bit encryption** checkbox.

5. Use Documentum Administrator or D2 Client to create a user with the same name as in Step 1 in your repository.

Note: You can run both D2 and Webtop on the same or different application servers while working with Kerberos SSO.

6. Perform the steps in [Adding Administrator Credentials to the D2 keystore for SSO](#), page 91
7. Perform the steps in [Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server](#) to configure the shiro.ini file.
8. Create and set the keytab:
 - a. In the command prompt, type the command `ktpass /pass <password> -out <computer name>.keytab -princ HTTP/<computer name>.<domain>@<DOMAIN> -crypto ALL +DumpSalt -ptype KRB5_NT_PRINCIPAL /mapOp set /mapUser <computer name>@<DOMAIN>`.
 - b. If you are using Microsoft Windows 2008 Active Directory, navigate to the **User Properties > Delegation** tab and select **Trust this user for delegation to any service (Kerberos only)**.
 - c. Copy the keytab file created to your application server machine.
9. Navigate to `webapps/D2/WEB-INF/classes/` and open `shiro.ini`. If `shiro.ini` does not exist, create a copy of `shiro_base.ini` and rename it as `shiro.ini`. Make the following changes to `shiro.ini`:
 - a. Find the line `X3-Kerberos.keyTabLocation` and append `=<location>`, where `<location>` is the path to the keytab you copied to the machine.
 - b. Add the lines:


```
[main]
X3-Kerberos=com.emc.x3.portal.server.filters.authc
.X3KerberosHttpAuthenticationFilter
X3-Kerberos.defaultRepository=<default repository>
X3-Kerberos.servicePrincipal=HTTP/<computer name>@<domain name>
X3-Kerberos.krbConfLocation=<path to KRB5.ini>
X3-Kerberos.keyTabLocation=<path to keytab file>
X3-Kerberos.debug=false
##Authentication type
/** = X3-Kerberos
```
 - c. Determine whether to use Kerberos SSO authentication for both application server and Documentum Server, or only for application server.

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

```
X3-Kerberos.endToEndSolution=true
```
10. Navigate to the **Windows** folder found in the operating system installation drive and open `KRB5.ini`.

Add the following lines:

```
[libdefaults]
default_realm=<DOMAIN>
```

```
[realms]
<DOMAIN> = {
kdc = <active directory server>.<domain>
}
```

11. If you are using Microsoft Windows 7 or Microsoft Windows 2008 R2:
 - a. Log in to the client machine with Administrator privileges.
 - b. Run `gpedit.msc`.
 - c. Navigate to **Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options > Network Security: Configure encryption types allowed for Kerberos**.
 - d. Select all options.
 - e. Click **OK**.
 - f. Restart the computer to enable the new group policy.
 - g. Log in to the client machine with the user account created in Step 1 to access the application.
12. If you are using the native ticket cache on a Microsoft Windows platform, the following exception may occur because Kerberos Ticket Granting Service is not exporting session keys:
javax.security.auth.login.LoginException: KrbException: KDC has no support for encryption type (14) – KDC has no support for encryption type
 - a. Navigate to the following registry path:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters
 - b. Set the following registry key:
Name: **allowtgtsessionkey**
Type: **REG_DWORD**
ValueL **0x01**
13. If you are using Kerberos on a Microsoft Windows 2008 R2 64-bit Kerberos Distribution Center machine, do not set **kdcUseRequestedEtypesForTickets** in the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc** registry path to 1. The configuration does not work after setting **kdcUseRequestedEtypesForTickets**.
14. Return to the instructions: [Instructions for Installing D2](#).

Configuring Kerberos for Multiple Domains

Setting up D2 with multi-domain Kerberos is same as single domain, with a few small differences. Multi-domain Kerberos is supported in the following configuration:

- The Documentum Server machine is in the parent domain, for example: `mdkparent.com`
- The Application Server and the D2 Client should be in the child domain, for example: `child1.mdkparent.com`

When creating the multiple domains, the following configuration should be used:

- The relationship between the parent and child domains should be - 2 way transitive trust.
- The relationship between child domains should be - 2 way transitive trust.

When generating the keytabs for the Application Server and mentioning the Application Server domain wherever applicable, the child domain (for example child1.parent.com) should be mentioned instead of a single domain (for example domain1.com).

Configuring TrustedReverseProxy for Various SSO Environments

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

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

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

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

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

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

5. For RSA Access Manager, set the following parameters:

- a. Specify the name of the HTTP request header which has the user's login name.

```
X3-TrustedReverseProxy.userParameterHeaderName=ct-remote-user
```

You can find the value for this property from RSA's `webagent.conf` "cleartrust.agent.exported_headers" setting. For example, `C:\Program Files (x86)\RSA\Access Manager Agent 5.0\Apache2.2\conf\webagent.conf`

- b. Specify a string that uniquely identifies the RSA Access Manager HTML login page.

```
X3-TrustedReverseProxy.logonForm=ctlogonform
```

Note: This setting is required only if the RSA Access Manager is configured to challenge users with an HTML login page.

To exclude all the resources (such as images, css, and so on) from the protected resources, add `/D2/resources/*` to the RSA's `webagent.conf` file. For example:

```
cleartrust.agent.url_exclusion_list=/cleartrust/images/*,/cleartrust
/styles/*,/D2/resources/*
```

- c. In case of SSO authentication for both application server and Documentum Server, specify the name of the authentication plugin on the Documentum Server.

```
X3-TrustedReverseProxy.ecsPlugIn=dm_rsa
```

- d. In case of SSO authentication for both application server and Documentum Server, specify the name of the RSA ClearTrust SSO cookie.

```
X3-TrustedReverseProxy.sessionCookieName=CTSESSION
```

When SSO is enabled, the RSA Web Agent creates this cookie and sets it in the browser. You can find the value for this property from RSA's `webagent.conf` "`cleartrust.agent.cookie_name`" setting. For example, `C:\Program Files (x86)\RSA\Access Manager Agent 5.0\Apache2.2\conf\webagent.conf`

- e. In case of SSO authentication for both application server and Documentum Server, specify the first optional argument available to the authentication plugins on the Documentum Server.

```
X3-TrustedReverseProxy.userArg1=RSA xxx.xxx.xxx.xxx 5608 2
```

The format is: `RSA <RSA-AccessMgr-host-ip> <RSA-AccessMgr-dispatcher-list-port> <security-level>` where, `security-level` represents the level of security used for network connections between the RSA Authorization Server and other components.

Three types of security levels are:

- clear = cleartext, no encryption.
- anon = anonymous SSL, SSL encryption only.
- auth = mutually authenticated SSL, SSL encryption with certificate-based authentication.

You can use 1 for clear, 2 for anon, and 3 for auth.

You can find the RSA Access Manager dispatcher list port from `dispatcher.conf`. For example, `C:\Program Files\RSA\Access Manager Servers 6.2\conf\dispatcher.conf`

D2 supports the establishment of multiple RSA Access Manager dispatchers, which can be specified using the following format:

```
RSA <dispatcher-ip> <dispatcher-port> <security-level>[,<dispatcher-ip> <dispatcher-port>
<security-level>]*
```

Example: `RSA 10.10.202.5 5608 2,10.10.202.6 5608 2,10.10.202.7 5608 2,10.10.202.8 5608 2`

6. For CA SiteMinder, set the following parameters:

- a. Specify the name of the HTTP request header which has the user's login name.

```
X3-TrustedReverseProxy.userParameterHeaderName=SM_USER
```

- b. Specify a string that uniquely identifies the CA SiteMinder HTML login page.

```
X3-TrustedReverseProxy.logonForm=login.fcc
```

Note: This setting is required only if the CA SiteMinder is configured to challenge users with an HTML login page.

- c. In case of SSO authentication for both application server and Documentum Server, specify the name of the authentication plugin on the Documentum Server.

```
X3-TrustedReverseProxy.ecsPlugIn=dm_netegrity
```

- d. In case of SSO authentication for both application server and Documentum Server, specify the name of the CA SiteMinder SSO cookie.

X3-TrustedReverseProxy.sessionCookieName=SMSESSION

7. For **IBM WebSEAL**, set the following parameters:

- a. Specify the name of the HTTP request header which has the user's login name.

X3-TrustedReverseProxy.userParameterHeaderName=iv-user

The WebSEAL junction for D2 should be created with the following command line:**server task <INSTANCE_NAME> create -t tcp -s -j -n -e utf8_uri -c iv-user -p <PORT> -h <HOST> / <JUNCTION_NAME>**

The following links contain useful information about configuring WebSEAL junctions and how the WebSEAL SSO login dialog works:[Configuring IV header authenticationSubmitting login form data directly to WebSEAL](#)

- b. Specify a string that uniquely identifies the WebSEAL HTML login page.

X3-TrustedReverseProxy.logonForm=pkmslogin.form

Note: This setting is required only if the WebSEAL is configured to challenge users with an HTML login page.

8. Perform the steps in [Configuring the Shiro.ini file for interoperability with D2-Config and the Documentum Method Server](#) to configure the shiro.ini file.

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

Adding Administrator Credentials to the D2 keystore for SSO

When D2 SSO has been configured with an implementation that uses DFC Principal Authentication, administrative credentials must be added to the D2 keystore.

1. Using the D2 keystore utility, add the following properties to the D2 keystore:

```
D2FS-trust.*.user=<administrator user>
D2FS-trust.*.password=<administrator password>
D2FS-trust.*.domain=<administrator user domain> [if relevant, not mandatory]
if the same credentials apply for all repositories of interest, or

D2FS-trust.repo1.user=< administrator user for repo1>
D2FS-trust.repo1password=< administrator password for repo1>
D2FS-trust.repo1.domain=< administrator user domain for repo1>
[if relevant, not mandatory]

D2FS-trust.repo2.user=< administrator user for repo2>
D2FS-trust.repo2password=< administrator password for repo2>
D2FS-trust.repo2.domain=< administrator user domain for repo2>
[if relevant, not mandatory]
```

Note: Property names are case sensitive.

Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server

When the administrator executes **Tools > Reload** options or **Tools > Refresh cache** in D2-Config, D2-Config or a D2Method running in JMS sends an HTTP request to each D2 application server listed in **Tools > Options > Client URLs**. If SSO is being used, however, there is no way for D2-Config or the D2Method to make a HTTP request in such a way that it will be authenticated by the D2 SSO authentication filter. For this reason, the `shiro.ini` file should be configured so that the corresponding servlet endpoints are unprotected by the D2 SSO authentication filter. To accomplish this, the following three lines should be added above the line that protects all other folders with the chosen type of SSO. For example, if Kerberos SSO is being used:

```
/servlet/ReloadOptions/* = anon/servlet/RefreshCache/* = anon/servlet
/LoadOnStartup/* = anon/** = X3-Kerberos
```

Although requests to these servlet endpoints will not be protected by the D2 SSO authentication filter it does not matter because requests to these servlet endpoints must include an encrypted admin login ticket on the URL, and the only clients capable of creating such URLs are D2-Config and the D2Method code that run in JMS.

The following D2 servlet endpoints need to bypass D2 SSO. The security is maintained because each of these servlet endpoints require that a valid login ticket is present in the URL.

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

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

Configuring the shiro.ini file for Custom SSO Integrations

Starting with D2 4.7, custom SSO integrations that use DFC Principal Mode authentication must explicitly declare in `shiro.ini` by setting `enableDFCPrincipalMode=true` for the custom SSO authentication filter.

1. Add relevant `D2FS-trust.*` properties to D2 Keystore.
2. Implement custom filter class (`CustomAuthenticationFilter`, for example). Here is a sample implementation:

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

```

```

        this.enabledDFCPrincipalMode = value;
    }

```

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

```

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

```

4. Rebuild and redeploy the custom filter to D2 server, then restart the server.

Configuring Security Assertion Markup Language (SAML) Support

D2 provides half-way Security Assertion Markup Language (SAML) support. D2 will validate the SAML token and use DFC Principal Mode which works by creating Documentum Server login tickets on behalf users who have been authenticated.

Note: Before installing D2 and configuring SAML support, you will need to set up your IdP properly, as described in the *IdP Installation Guide*

Note: Please refer to the core REST documentation for information on setting up the SAML environment for REST APIs.

1. It is crucial that the machine clock of the IdP, service provider, and client machines all be in sync. Once all the machines are synched using NTP, the clocks might still have a slight difference. You need to allow for clock skew. For example, in the ADFS IdP environment, you can use the following command to get/set the clock skew or sign the authentication request:

```
Get-ADFSRelyingPartyTrust -Name "relying_party"
```

2. Run the following command to set the **NotBeforeSkew** property of the relying party trust settings. One minute is more than enough to cover the clock skew:

```
Set-ADFSRelyingPartyTrust -Targetname "relying_party"-NotBeforeSkew 1
```

NotBeforeSkew indicates that the SAML assertion is not valid before a particular time. But, if the SP's clock is slower than the ADFS, the SP could receive the SAML assertion when the SP machine clock is before the "NotBefore" time.

3. When you are configuring the relying party trust, you have the choice of making the service provider signing of the authentication request mandatory or optional. For example, run the following command to display the relying party trust settings in an ADFS IdP environment:

```
Get-ADFSRelyingPartyTrust -Name "relying_party"
```

4. Run the following command with the **SignedSamlRequestsRequired** property. Setting it to **true** will make the signing mandatory, **false** is optional.

```
Set-ADFSRelyingPartyTrust -Targetname "relying_party" -
SignedSamlRequestsRequired <true/false>
```

If the signing of the SAML authentication request is mandatory, then you will need to provide the certificate used to verify the service provider signature to IdP.

5. Perform the steps in *Adding Administrator Credentials to the D2 Keystore* for SSO as noted below to enable SSO half-way support.

- Navigate to `webapps/D2/WEB-INF/classes` and open `shiro.ini`. If `shiro.ini` does not exist, create a copy of `shiro-base.ini` and rename it to `shiro.ini`.
- Edit the general SAML parameters in `shiro.ini` file with the details shown below:

```
X3-SAML=com.emc.x3.portal.server.filters.authc.X3SAMLHttpAuthenticationFilter
X3-SAML.defaultRepository=<default repository>
```

Here is the Identity Provider (IdP) URL that will generate the SAML Assertion:

```
X3-SAML.idpUrl=https://d2-adfs.d2saml.com/adfs/ls
```

This is the URL where the SAML Assertion will be posted back. It needs to be configured in the Endpoint tab of relying trust party.

```
X3-SAML.assertionConsumerServiceUrl=https://d2-appdev.d2saml.com:8443/D2/
```

The value of the issuer is the relying party identifier:

```
X3-SAML.issuer=TestSAML
```

IdP signs the SAML response. Specify the absolute path of the IdP certificate used to verify if the SAML response is actually coming from the IdP:

```
X3-SAML.idpTokenSigningCertificate=C:/SAML/certs/idp-token-signing.cer
```

D2 supports sending signed and unsigned SAML authentication requests. If you want the SAML authentication request to be signed, then you must provide the absolute path of the Java key store file which contains quantities used to sign the request. If you don't specify the **jksLocation**, then the SAML request will not be signed, and if the IdP requires that authentication requests be signed, the authentication request will fail.:

```
X3-SAML.jksLocation=C:/saml/certs/serviceprovider.keystore
```

The following property is used only if **jksLocation** is defined. A service provider identifier may be associated with each D2 app server, and this identifier is used to get the jks password, jks key entry alias and jks key entry password properties from the D2 keystore that are needed in order to sign a SAML authentication request. Use the **D2KeyStoreUtil** command line utility to add such properties to the D2 keystore. For a specific service provider identifier, add the following properties:

```
jks.<service provider identifier>.password
jks.<service provider identifier>.entry.alias
jks.<service provider identifier>.entry.password
```

To define default values for these properties, add the following properties:

```
jks.*.password
jks.*.entry.alias
jks.*.entry.password
```

If a property is not defined for a specific service provider identifier, the corresponding default value, if defined, will be used instead. This arrangement allows you to configure some or all D2 app servers (for example, service providers) with their own java keystore password, **entry.alias** and **entry.password** property values, or to use a common set of property values valid for all D2 app servers.

X3-SAML.serviceProviderIdentifier=d2-app-svr

Installing Advanced Publishing Services

Advanced Publishing Services Plug-in

Advanced Publishing Services is a D2 plug-in that adds the ability to publish Virtual Documents in D2. The plug-in allows publishing virtual documents in two ways:

- Paper publishing: to publish PDF files.
- Electronic publishing: to publish files into a folder structure.

Deploying Advanced Publishing Services Automatically

The following installation contains the steps for automatically deploying the DAR file and then installing the Advanced Publishing plug-in library files on both the Documentum Server and application server:

1. Stop the Application Server or Documentum Java Method Server or both.
2. Run the `AdvancedPublishing-Install-<version>.jar` using the following command:

```
java -jar AdvancedPublishing-Install.<version>.jar
```
3. Select the installation path for the plug-in. Click **Next**.
4. Select **Content Server** or **Application Server** or both, depending on where you want to run the plug-in installation. Click **Next**.
5. If you have selected Content Server:
 - a. Type the Content Server install owner's name. Click **Next**.
 - b. Type the repositories name (separated by comma) for which you would like to install D2. Click **Next**.
6. If you are installing the plug-in for application server, specify the application server location where you have already deployed the D2 or D2 Config. For example, **webapps** in Apache Tomcat. Click **Next**.
7. Read the extraction summary and click **Next**, then click **Done**.

Example 10-1.

Installing the Advanced Publishing Services Plug-in

The following installation contains the steps for deploying the DAR file and then installing the Advanced Publishing Services plug-in library files on both the Documentum Server and application server:

1. Run the AdvancedPublishing plug-in installer (if not done already) as described in the [Deploying Advanced Publishing Services Automatically, page 97](#) section.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR Installer shipped with Documentum Composer, **dardeployer.exe**, and fill out the form as described in the following table:

Note: The DAR installer is located in the directory you selected while running the plug-in installer.

Field	Description
DAR	Select P2-DAR.dar
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually dmadmin. The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the nodmadmin.installparam file if the Documentum Server installation owner is not named dmadmin, as described in Step 4.

4. If the Documentum Server installation owner is not dmadmin:
 - a. Create a file in a text editor and save it as **nodmadmin.installparam**.
 - b. Add the following lines:

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

where `Administrator` is the name of the account owner for the installation.

- c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.
7. Stop Documentum Server, JMS services, and web application services.
8. Follow the instructions on both the Documentum Server and the application for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing Advanced Publishing Services Libraries on Microsoft Windows, page 100	Installing Advanced Publishing Services Libraries on a Linux Environment, page 99

9. Restart the Documentum Server and the application server.

Installing Advanced Publishing Services Libraries on a Linux Environment

1. On the Documentum Server, copy `P2-API.jar` from the extraction folder to the folder depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<code>\$DOCUMENTUM/<Java Method Server>/server/DctmServer_MethodServer/deployments/ServerApps.ear/lib</code>	<code>\$DOCUMENTUM/<Java Method Server>/server/DctmServer_MethodServer/deploy/ServerApps.ear/lib</code>

2. On the application server:
 - a. Copy `P2-API.jar` from the extraction folder to the `D2-Config/WEB-INF/lib` folder.
 - b. Copy `P2-Plugin.jar` from the extraction folder to the `D2-Config/WEB-INF/classes/plugins` folder.
 - c. Copy `P2-API.jar` and `P2-Plugin.jar` from the extraction folder to the `D2/WEB-INF/lib` folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is `<install path to web application server>/webapps/D2-Config/WEB-INF/classes`
 - b. Open for editing `D2-Config.properties`.
 - c. Add and set the value of `plugin_x=` to the path for `P2-Plugin.jar`.
Use forward slashes for the file path. For example:
 - To use an absolute path: `<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/P2-Plugin.jar`
 - To use a relative path: `/plugins/P2-Plugin.jar`

x equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**

- d. If you have installed other plug-ins such as C2 and O2, specify the plug-in order in D2 by modifying the `pluginsOrder` property in the `D2FS.properties` file:

pluginsOrder=AdvancedPublishing,C2,O2.

Note: Ensure that the Advanced Publishing is first in the plug-ins list.

4. Return to the Advanced Publishing Services installation instructions: [Installing the Advanced Publishing Services Plug-in, page 98.](#)

Installing Advanced Publishing Services Libraries on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On the Documentum Server, copy **P2-API.jar** from the extraction folder to the folder depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deployments\ServerApps.ear\lib	<install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\ServerApps.ear\lib

2. On the application server:
- Copy **P2-API.jar** from the extraction folder to the `D2-Config\WEB-INF\lib` folder.
 - Copy **P2-Plugin.jar** from the extraction folder to the `D2-Config\WEB-INF\classes\plugins` folder.
 - Copy **P2-API.jar** and **P2-Plugin.jar** from the extraction folder to the `D2\WEB-INF\lib` folder.

3. Activate the plug-in:

- Navigate to the location of your D2 Config configuration files.
The default location is `<install path to web application server>\webapps\D2-Config\WEB-INF\classes`
- Open for editing `D2-Config.properties`.
- Add and set the value of `plugin_x=` to the path for `P2-Plugin.jar`.
Use forward slashes for the file path. For example:
 - To use an absolute path: `<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/P2-Plugin.jar`
 - To use a relative path: `/plugins/P2-Plugin.jar`

x equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**

- d. If you have installed other plug-ins such as C2 and O2, specify the plug-in order in D2 by modifying the `pluginsOrder` property in the `D2FS.properties` file:

pluginsOrder=AdvancedPublishing,C2,O2.

Note: Ensure that the Advanced Publishing is first in the plug-ins list.

4. Return to the Advanced Publishing Services installation instructions: [Installing the Advanced Publishing Services Plug-in, page 98](#).

Installing D2-Link

The D2 Applet provides a D2-Link application that is installed on the client machine. The D2-Link application is required for the Hyperlink feature. The D2-Link application is required to view the available hyperlinks and also to open a target document from a hyperlink in the source document. The D2-Link application only works in java mode.

Note: Before installing the D2-Link, make sure the `JAVA_HOME` environment variable is set in the client machine.

The `D2-Link.jar` is installed automatically every time the D2 Applet is initialized. The D2 Applet checks whether `X3Applet.properties` exists under `<path to the user's home directory>/Documentum/D2`.

The D2 Applet checks, whether the operating system is Windows, and the `d2link.automated.install` property in `X3Applet.properties` is true.

The D2 Applet also verifies whether the `D2-Link.jar` file exists in the folder `<path to the user's home directory>/Documentum/D2`. When it does not exist, the D2 Applet downloads the jar file from the D2 application server, saves the file in this location, and registers the `d2link` protocol with Windows by writing to the HKCU hive of the Windows registry.

Note: The default value for `d2link.automated.install` in `X3Applet.properties` is true, which means that the `D2-Link.jar` is automatically installed and registered. When a newer version of `D2-Link.jar` becomes available, D2 administrators must inform their user community to remove the `D2-Link.jar` from their `<path to the user's home directory>/Documentum/D2` folder to download and register the newer version.

If the above mentioned installation and registration mechanism does not work, you can download and register the `D2-Link.jar` by using the following steps:

1. Type the following URL into the browser address field:
`<protocol>://<server-address>:<port no>/D2/applet/D2-Link.jar`
 For example: `http://d2server.acme.com:8080/D2/applet/D2-Link.jar`
2. Save the `D2-Link.jar` to a local folder location. For example, `<path to the user's home directory>/Documentum/D2`.
3. Open a command prompt window and change to the directory folder where the `D2-Link.jar` file was downloaded and saved.

4. Run the following command:

```
java -jar D2-Link.jar
```

The `d2link` protocol is successfully registered.

Troubleshooting the Installation

If you run into issues when installing, it is recommended to delete temporary Internet files.

Installing Advanced Publishing Services Manager

Advanced Publishing Services Manager

Advanced Publishing Services Manager, is a Java application that is used to manage and process publishing in D2. It can be run either as a Swing graphical interface or in the Console mode

Advanced Publishing Services Manager can be used:

- By end users, for processing Advanced Publishing Services publishing asynchronously.
- By administrators, to manage queues from inside D2 Client (provided Advanced Publishing Services is installed).

Pre-requisites for Advanced Publishing Services Manager

Install and get the D2 and Advanced Publishing Services API on the Documentum Server. If you want to install the Advanced Publishing Manager on a server other than Documentum Server, install Documentum Foundation Classes (DFC). The DFC version varies based on your Documentum Server version. The *OpenText Documentum D2 Release Notes* provides more information on the supported API versions and system requirements.

Verify the Documentum Server for D2-API mandatory libraries:

1. Ensure that D2-API is already installed on the Documentum Server. On a server other than Documentum Server, manually copy the D2-API library files (*.jar) from the Documentum Server to the other server where you want to run the Advanced Publishing Manager. Default D2-API location in Documentum Server is : C:\Program Files\EMC\D2
2. Ensure that Advanced Publishing Services are already installed on the Documentum Server. On a server other than Documentum Server, manually copy Advanced Publishing Services API libraries (P2-API.jar) from the Documentum Server.
3. Download and extract the corresponding DFS SDK. The DFS SDK version varies based on the Documentum Server version.
4. In the CLASSPATH variable, add dctm.jar and the **Documentum/config/** repository. For example, typical paths are <documentum install folder>/dctm.jar and <documentum install folder>/Config/

5. Set the CLASSPATH for the D2 and Advanced Publishing Services APIs:
 - a. For D2 API: <install path of D2>/D2.jar; <install path of D2>/D2FS4DCTM-API.jar;install path of D2>D2FS-Generated.jar
 - b. For DFS SDK: <install path of DFS SDK>/lib/java/emc-dfs-rt.jar install path of DFS SDK>/lib/java/emc-dfs-rt-remote.jar
 - c. For Advanced Publishing Services API: <install path of Advanced Publishing Services>/P2-API.jar
6. Set the following variables before beginning your installation:
 - **<documentum_folder>**: directory of the Documentum environment (references the files in the **Config** folder).
 - **<dfc_folder>**: directory of the Documentum executable installation (references the files in the **Shared** folder).

Note: It is recommended that you should use Advanced Publishing Manager on a server other than Documentum Server. The reasons being performance slowdown depending on the volume of documents being published.

Installing Advanced Publishing Services Manager

1. Copy the AdvancedPublishingManager-Install.jar file to the target server.
 - a. On Windows, right-click on the file, select **Open with > Java™**.
 - b. On Linux, run the installer from an **XWindows** interface using the Documentum installation owner account. Open an xterm and run the installer with the command-line **java -jar AdvancedPublishingManager-Install.jar**
2. In the Advanced Publishing Services Manager installation wizard, click **Next**.
3. Select P2 Manager.
4. Browse and select the target directory. The default location is:
 - a. On Windows, C:\Program Files\EMC\AdvancedPublishingManager
 - b. On Linux,/opt/EMC/AdvancedPublishingManager
5. Click **Next** and then **OK** to create the target directory.
6. Click **Next** after the installation progress is completed.
7. Note down the post installation tasks necessary for completing Advanced Publishing Services Manager installation.
8. Click **Done**.

Configuring Advanced Publishing Services Manager to run in Console Mode

To run Advanced Publishing Services Manager in the console mode:

1. Ensure that the CLASSPATH contains the following references to run Advanced Publishing Services Manager through the command line:
 - a. **dfc.jar** and the Documentum Foundation Classes (DFC) configuration folder to allow running DFC-based programs.
 - b. **D2.jar** library to allow running D2-based programs.
 - c. **P2-API.jar** to include Advanced Publishing Services libraries.
2. Navigate to the installation folder of Advanced Publishing Services Manager and open **P2Manager_config.xml** in a text editor. Replace values such as the login and the encrypted password.
To encrypt a password, add **D2.jar** to the system CLASSPATH and use the command `java.com.emc.d2.api.utils.GetCryptedPassword`
3. After saving the **P2Manager_config.xml** file, launch Advanced Publishing Services Manager in console mode using one of the following methods:
 - Run the `launchP2ManagerConsole.exe`
 - Run the `launchP2ManagerConsole.bat` batch file.
 - Execute `java -cp $CLASSPATH:./lib/*:./bin com.emc.p2.manager.Main -console`

Viewing log information

By default, Advanced Publishing Services Manager logs are located in the directory:

`C:/logs/P2Manager.log`. A `logback.xml` file is also available under the Advanced Publishing Services Manager install folder. This `logback.xml` file contains when, where, and how application logging events should be written and should be modified if required. Advanced Publishing Services Manager uses a logback logging system, based on an abstraction library (slf4j). Using this abstraction, the logging system can be changed based on which library is packaged (log4j, jul, jcl, logback) with your web application.

Advanced Publishing Services Manager logs can be very verbose, especially in debug mode. In the `logback.xml`, you can specify an option to display publishing logs in a dedicated file:

```
<appender class="ch.qos.logback.core.rolling.RollingFileAppender"
name="PubliFileAppender">
<file>C:/logs/P2ManagerPubli.log</file>
<!--<bufferedIO>true</bufferedIO-->
<append>true</append>
<filter class="ch.qos.logback.classic.filter.ThresholdFilter">
<level>TRACE</level>
</filter>

<rollingpolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
<fileNamePattern>C:/logs/P2Manager-%d{yyyy-MM-dd}Publi.log.zip</fileNamePattern>
<MaxHistory>5</MaxHistory>
</rollingPolicy>
```

```
<layout class="ch.qos.logback.classic.PatternLayout">
<pattern>%-27(%date{yyyy-MM-dd HH:mm:ss} [%-5level]) -
% logger{64}: %message%n</pattern>
</layout>
</appender>

<logger name="com.emc.p2.manager.thread.P2ManagerThread">
<level value="TRACE"/>
<appender-ref ref="PubliFileAppender"/>
</logger>
```

You can filter on a pattern if “debugFilter” is set to a correct and single value. For example:

```
<filter class="ch.qos.logback.core.filter.EvaluatorFilter">
<evaluator name="messaging">
<expression>logger.getName().contains("${debugFilter:-Messaging}") ||
event.getFormattedMessage().contains("${debugFilter:-Messaging}")</expression>
<evaluator>
<OnMismatch>NEUTRAL</OnMismatch>
<OnMatch>DENY</OnMatch>
</filter>
```

You can find the lines in the current logback.xml file. Ensure that only one filter is activated for one appender definition. Each time the word Messaging appears in the logging event or logger name, that line will be omitted from the appender.

Wherever you find the <level value="warn"/>command, it indicates that in a logger context, the level of threshold is **warn**. For example, if you find the following: mechanism: <level value="\$ {debug:-warn}"/>

```
logger name="com.emc.d2.web.servlets.D2Context"
<level value="warn"/>
<appender-ref ref="RootFileAppender"/>
</logger>
```

To create or change a logger, you need the entire class name (normally found from the log file). You also need to change the activated threshold level if needed (the default level is ancestor) and set the appender reference.

To create or change an appender, and to merge log files or set a different mechanism:

```
<appender class="ch.qos.logback.core.ConsoleAppender"
name="RootConsoleAppender">
</layout>
<pattern>%27(%date{yyyy-MM-dd HH:mm:ss} [%-5level])
%logger{64}: %message%n</pattern>
</layout>
</appender>
```

Depending on the class used for the appender (file, console, DB and so on) different sets of options are available. The layout determines the way the log message information is displayed into the appender. Using the file appender, you can set up the file name and location, buffering, and filtering on a log level. You can set up one appender file per type of log (debug, warn, info).

Note: Changing appender options can consume resources and affect application server performance.

Running Multiple Instances of Advanced Publishing Services Manager

Only a single instance of an Advanced Publishing Services Manager installation can be run at a time on a machine. This is to preserve publishing queue consistency.

To allow a single machine to run multiple instances of Advanced Publishing Services Manager, you will have to make as many as installations of Advanced Publishing Services Manager as you need instances.

For example, if you plan to have two instances of Advanced Publishing Services on your server, you will need to install each instance in a different folder. Each Advanced Publishing Services Manager installation will have its unique settings and ID. This unique ID is automatically generated using Machine Name. If you require more than one instance, then it is recommended that you use different log files.

When planning for multiple instances of Advanced Publishing Services Manager, watching the same queue, it is recommended to put higher values in the **Delay between threads** field.

Session Pooling

To configure the pooling session on the application server:

Ensure that the `dfc.properties` file includes the following lines: `[DMAPI_CONFIGURATION]`
`connect_pooling_enabled=T connect_recycle_interval=900`

This enables the pool session and sets the recycle interval to 900, that is, 15 minutes. When a session has spent this interval of time, it is deleted and created again.

Launching Advanced Publishing Services Manager on Windows

Advanced Publishing Services can be scheduled to launch automatically at specific times. The schedule can be specified by using Scheduled Tasks to execute the `launchP2ManagerConsole.bat` file. Using Scheduled Tasks makes it possible to use the bat file as a Windows service using the Standard Windows Service creation process. For information about creating a user-defined service, refer to documentation available on the [Microsoft Web site](#).

However, the bat file at times may launch a Java process that does not stop even when the Windows service is stopped.

Installing C2

Deploying C2 Automatically

The C2 plug-in adds the Portable Document Format (PDF) control capabilities to D2 Client. The following installation contains the steps for automatically deploying the DAR file and then installing the C2 plug-in library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the C2 plug-in. If C2 plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the `C2-<version>-install.jar`.
2. Select the installation path for the plug-in. Click **Next**.
3. Select **Content Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Content Server install owner's name. Click **Next**.
5. Type the repositories name (separated by comma) for which you would like to install D2. Click **Next**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy D2 Config.
7. Read the extraction summary and click **Next**, then click **Done**.

Installing the C2 Plug-in

The following installation contains the steps for deploying the DAR file and then installing the C2 plug-in library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the C2 plug-in. If C2 plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure

1. Run the C2 plug-in installer (if not done already) as described in the [Deploying C2 Automatically, page 109](#) section.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Note: The DAR installer is located in the directory you selected while running the plug-in installer.

Field	Description
DAR	Select C2-DAR.dar
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually dmadmin . The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the nodmadmin.installparam file if the Documentum Server installation owner is not named dmadmin , as described in Step 4.

4. If the Documentum Server installation owner is not **dmadmin**:
 - a. Create a file in a text editor and save it as **nodmadmin.installparam**.
 - b. Add the following lines:

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

where **Administrator** is the name of the account owner for the installation.
 - c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the **nodmadmin.installparam** you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.
7. Stop Documentum Server, JMS services, and web application services.
8. Follow the instructions on both the Documentum Server and the application for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing C2 Libraries on Microsoft Windows	Installing C2 Libraries on a Linux Environment

9. Restart the Documentum Server and the application server.

Installing C2 Libraries on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On the Documentum Server, copy **C2-API.jar** from the extraction folder to the folder depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<pre><install path of Documentum>\<Java Method Server>\server\DctmServer _MethodServer\deployments _ServerApps.ear\lib</pre>	<pre><install path of Documentum>\<Java Method Server>\server\DctmServer _MethodServer\deploy\ServerApps .ear\lib</pre>
and	and
<pre><install path of Documentum>\<Java Method Server>\server\DctmServer _MethodServer\deployments\bpm.ear \lib</pre>	<pre><install path of Documentum>\<Java Method Server>\server\DctmServer _MethodServer\deploy\bpm.ear\lib \lib</pre>

2. On the application server:
 - a. Copy **C2-API.jar** from the extraction folder to the **D2-Config\WEB-INF\lib** folder.
 - b. Copy **C2-Plugin.jar** from the extraction folder to the **D2-Config\WEB-INF\classes\plugins** folder.
 - c. Copy **C2-API.jar** and **C2-Plugin.jar** from the extraction folder to the **D2\WEB-INF\lib** folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is **<install path to web application server>\webapps\D2-Config\WEB-INF\classes**
 - b. Open for editing **D2-Config.properties**.
 - c. Add and set the value of **plugin_x=** to the path for **C2-Plugin.jar**.
Use forward slashes for the file path. For example:
 - To use an absolute path: **<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/C2-Plugin.jar**
 - To use a relative path: **/plugins/C2-Plugin.jar****x** equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**
4. Return to the C2 installation instructions: [C2 Overview and Roadmap](#).

Installing C2 Libraries on a Linux Environment

1. On the Documentum Server, copy **C2-API.jar** from the extraction folder to the following folders depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<pre>\$DOCUMENTUM/<Java Method Server> /server/DctmServer_MethodServer /deployments/ServerApps.ear/lib</pre>	<pre>\$DOCUMENTUM/<Java Method Server> /server/DctmServer_MethodServer /deploy/ServerApps.ear/lib</pre>
and	and
<pre>\$DOCUMENTUM/<Java Method Server> /server/DctmServer_MethodServer /deployments/bpm.ear/lib</pre>	<pre>\$DOCUMENTUM/<Java Method Server>/server/DctmServer _MethodServer/deploy/bpm.ear/lib</pre>

2. On the application server:
 - a. Copy **C2-API.jar** from the extraction folder to the **D2-Config/WEB-INF/lib** folder.
 - b. Copy **C2-Plugin.jar** from the extraction folder to the **D2-Config/WEB-INF/classes/plugins** folder.
 - c. Copy **C2-API.jar** and **C2-Plugin.jar** from the extraction folder to the **D2/WEB-INF/lib** folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is **<install path to web application server>/webapps/D2-Config/WEB-INF/classes**
 - b. Open for editing **D2-Config.properties**.
 - c. Add and set the value of **plugin_x=** to the path for **C2-Plugin.jar**.
Use forward slashes for the file path. For example:
 - To use an absolute path: **<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/C2-Plugin.jar**
 - To use a relative path: **/plugins/C2-Plugin.jar****x** equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**
4. Return to the C2 installation instructions: [C2 Overview and Roadmap](#).

Installing O2

Deploying O2 Automatically

The O2 plug-in adds the ability to manage transferring properties between D2 and Microsoft Office documents.

The following installation contains the steps for automatically deploying the DAR file and then installing the O2 plug-in library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the O2 plug-in. If O2 plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the `O2-<version>-install.jar`.
2. Select the installation path for the plug-in. Click **Next**.
3. Select **Content Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Content Server install owner's name. Click **Next**.
5. Type the repositories name (separated by comma) for which you would like to install D2. Click **Next**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy D2 Config.
7. Read the extraction summary and click **Next**, then click **Done**.

Example 13-1.

Installing the O2 Plugin

The following installation contains the steps for deploying the DAR file and then installing the O2 plugin library files on both the Documentum Server and application server:

Note: D2 with the O2 plugin installed causes all RPS jobs to fail and JMS to run out of memory. This is caused by a JAXB conflict with the `geronimo-stax-api_1.0_spec-1.0.jar` file. This JAR file is required for D2 with O2 plugin functions but when present on JMS causes all RPS to fail.

Note: If directed, D2 Core installation automatically deploys the O2 plugin. If O2 plugin was not indicated during the D2 Core installation or if you want to update the C2 plugin, follow this procedure.

1. Run the O2 plugin installer (if not done already) as described in the [Deploying O2 Automatically, page 113](#) section.

2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Note: The DAR installer is located in the directory you selected while running the plugin installer.

Field	Description
DAR	Select O2-DAR.dar
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually <code>dmadmin</code> . The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the <code>nodmadmin.installparam</code> file if the Documentum Server installation owner is not named <code>dmadmin</code> , as described in Step 4.

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


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

 where `Administrator` is the name of the account owner for the installation.
 - c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the `nodmadmin.installparam` you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.
7. Stop Documentum Server, JMS services, and web application services.
8. Follow the instructions on both the Documentum Server and the application for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing O2 Libraries on Microsoft Windows	Installing O2 Libraries on a Linux Environment

9. Restart the Documentum Server and the application server.

Installing O2 Libraries on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On the Documentum Server, copy **O2-API.jar** from the extraction folder to the folder on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<code><install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deployments\ServerApps.ear\lib</code>	<code><install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\ServerApps.ear\lib</code>

2. On the application server:
 - a. Copy **O2-API.jar** from the extraction folder to the **D2-Config\WEB-INF\lib** folder.
 - b. Copy **O2-Plugin.jar** from the extraction folder to the **D2-Config\WEB-INF\classes\plugins** folder.
 - c. Copy **O2-API.jar** and **O2-Plugin.jar** from the extraction folder to the **D2\WEB-INF\lib** folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is `<install path to web application server>\webapps\D2-Config\WEB-INF\classes`
 - b. Open for editing **D2-Config.properties**
 - c. Add and set the value of **plugin_x=** to the path for **O2-Plugin.jar**.
Use forward slashes for the file path. For example:
 - To use an absolute path: `<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/O2-Plugin.jar`
 - To use a relative path: `/plugins/O2-Plugin.jar`**x** equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**
4. Return to the O2 installation instructions: [O2 Overview and Roadmap](#).

Installing O2 Libraries on a Linux Environment

1. On the Documentum Server, copy **O2-API.jar** from the extraction folder to the folder on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
\$DOCUMENTUM/<Java Method Server> /server/DctmServer_MethodServer /deployments/ServerApps.ear/lib	\$DOCUMENTUM/<Java Method Server> /server/DctmServer_MethodServer /deploy/ServerApps.ear/lib

2. On the application server:
 - a. Copy **O2-API.jar** from the extraction folder to the **D2-Config/WEB-INF/lib** folder.
 - b. Copy **O2-Plugin.jar** from the extraction folder to the **D2-Config/WEB-INF/classes/plugins** folder.
 - c. Copy **O2-API.jar** and **O2-Plugin.jar** from the extraction folder to the **D2/WEB-INF/lib** folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is **<install path to web application server>/webapps/D2-Config/WEB-INF/classes**
 - b. Open for editing **D2-Config.properties**
 - c. Add and set the value of **plugin_x=** to the path for **O2-Plugin.jar**.
Use forward slashes for the file path. For example:
 - To use an absolute path: **<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/O2-Plugin.jar**
 - To use a relative path: **/plugins/O2-Plugin.jar****x** equals the number of previous plug-in plus one. If no other plug-in is installed, use **plugin_1**
4. Return to the O2 installation instructions: [O2 Overview and Roadmap](#).

Installing D2-Bin

Deploying D2–Bin Automatically

The D2-Bin plug-in adds recycling bin capabilities to D2 Client. The following installation contains the steps for automatically deploying the DAR file and then installing the D2-Bin plug-in library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the D2–Bin plug-in. If D2–Bin plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the `D2-Bin-<version>-install.jar`.
2. Select the installation path for the plug-in. Click **Next**.
3. Select **Content Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Content Server install owner's name. Click **Next**.
5. Type the repositories name (separated by comma) for which you would like to install D2. Click **Next**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy D2 Config.
7. Read the extraction summary and click **Next**, then click **Done**.

Installing the D2-Bin Plug-in

The following installation contains the steps for deploying the DAR file and then installing the D2-Bin plug-in library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the D2–Bin plug-in. If D2–Bin plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the D2–Bin plug-in installer (if not done already) as described in the [Deploying D2–Bin Automatically, page 117](#) section.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Note: The DAR installer is located in the directory you selected while running the plug-in installer.

Field	Description
DAR	Select D2-Bin-DAR.dar
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually dmadmin . The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the nodmadmin.installparam file if the Documentum Server installation owner is not named dmadmin , as described in Step 4.

4. If the Documentum Server installation owner is not **dmadmin**:
 - a. Create a file in a text editor and save it as **nodmadmin.installparam**.
 - b. Add the following lines:

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

where **Administrator** is the name of the account owner for the installation.
 - c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the **nodmadmin.installparam** you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.
7. Stop the Documentum Server, JMS services, and web application services.
8. Follow the instructions on both the Documentum Server and the application for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing D2-Bin on Microsoft Windows	Installing D2-Bin on a Linux Environment

9. Restart the Documentum Server, JMS services, and the application server.

Installing D2-Bin on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On the Documentum Server, copy **D2-Bin-API.jar** from the extraction folder to the folder depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deployments_ServerApps.ear\lib	<install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deploy\ServerApps.ear\lib

2. On the application server:
 - a. Copy **D2-Bin-API.jar** from the extraction folder to the **D2-Config\WEB-INF\lib** folder.
 - b. Copy **D2-Bin-Plugin.jar** from the extraction folder to the **D2-Config\WEB-INF\classes\plugins** folder.
 - c. Copy **D2-Bin-API.jar** and **D2-Bin-Plugin.jar** from the extraction folder to the **D2\WEB-INF\lib** folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is <install path to web application server>\webapps\D2-Config\WEB-INF\classes
 - b. Open for editing **D2-Config.properties**
 - c. Add the line **plugin_x=<D2 Config installation path>\D2-Config\WEB-INF\classes\plugins\D2-Bin-Plugin.jar**
If no other plug-in is installed, x equals one.
Otherwise, x equals the number of previous plug-in plus one.
4. Return to the D2-Bin installation instructions: [D2-Bin Overview and Roadmap](#).

Installing D2-Bin on a Linux Environment

1. On the Documentum Server, copy **D2-Bin-API.jar** from the extraction folder to the folder depending on your Documentum Server version, as described below:

Documentum Server 7.1 and later	Documentum Server 7.0
<install path of Documentum>\<Java Method Server>\server\DctmServer_MethodServer\deployments_ServerApps.ear\lib	\$DOCUMENTUM/<Java Method Server>/server/DctmServer_MethodServer/deploy/ServerApps.ear/lib

2. On the application server:
 - a. Copy **D2-Bin-API.jar** from the extraction folder to the **D2-Config/WEB-INF/lib** folder.

- b. Copy **D2-Bin-Plugin.jar** from the extraction folder to the **D2-Config/WEB-INF/classes/plugins** folder.
 - c. Copy **D2-Bin-API.jar** and **D2-Bin-Plugin.jar** from the extraction folder to the **D2/WEB-INF/lib** folder.
 3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is **<install path to web application server>/webapps/D2-Config/WEB-INF/classes**
 - b. Open for editing **D2-Config.properties**
 - c. Add the line **plugin_x=<D2 Config installation path>\D2-Config\WEB-INF\classes\plugins\D2-Bin-Plugin.jar**
If no other plugin is installed, x equals one.
Otherwise, x equals the number of previous plugin plus one.
 4. Return to the D2-Bin installation instructions: [D2-Bin Overview and Roadmap](#).

Installing D2 RPS Connector

Deploying D2 RPS Connector Automatically

The D2 RPS Connector plug-in adds the ability for D2 to consume Documentum Retention Policies and a set of configuration modules to D2 Config. To use the retention policies in D2, you must configure policies using Documentum Retention Policy Services. The *OpenText Documentum Records Client Administration and User Guide* contains further information. The following installation contains the steps for automatically deploying the DAR file and then installing the D2 RPS Connector plugin library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the D2 RPS Connector plug-in. If D2 RPS Connector plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the `D2-RPS-<version>-install.jar`.
2. Select the installation path for the plug-in. Click **Next**.
3. Select **Content Server** and/or **Application Server** where you want to run the plug-in installation. Click **Next**.
4. Type the Content Server install owner's name. Click **Next**.
5. Type the repositories name (separated by comma) for which you would like to install D2. Click **Next**.
6. If you are installing the plug-in for application server, specify the application server location where you want to deploy D2 Config.
7. Read the extraction summary and click **Next**, then click **Done**.

Installing the D2 RPS Connector Plug-in

The following installation contains the steps for deploying the DAR file and then installing the D2 RPS Connector plugin library files on both the Documentum Server and application server:

Note: If directed, D2 Core installation automatically deploys the D2 RPS Connector plug-in. If D2 RPS Connector plug-in was not indicated during the D2 Core installation or if you want to update the C2 plug-in, follow this procedure.

1. Run the D2 RPS Connector plug-in installer (if not done already) as described in the [Deploying D2 RPS Connector Automatically, page 121](#) section.
2. Ensure that the Docbroker and the target repository are running.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Note: The DAR installer is located in the directory you selected while running the plug-in installer.

Field	Description
DAR	Select D2-RPS.dar
Docbroker Details	Select the target Docbroker and port. Click Connect .
Repository Details	Select the repository with the Documentum Server installation owner account, usually dmadmin . The installation owner account must have Super User privileges in the repository when deploying the dar files. Type the login and password for the owner account.
Input File	Select the nodmadmin.installparam file if the Documentum Server installation owner is not named dmadmin , as described in Step 4.

4. If the Documentum Server installation owner is not **dmadmin**:
 - a. Create a file in a text editor and save it as **nodmadmin.installparam**.
 - b. Add the following lines:


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

 where **Administrator** is the name of the account owner for the installation.
 - c. Under **DAR Details**, click **Browse** next to **Input File**, and locate and select the **nodmadmin.installparam** you created.
5. Click **Install**.
6. Click **Recent DAR install log files** to review log files.
7. Stop Documentum Server, JMS services, and web application services.
8. Follow the instructions on both the Documentum Server and the application for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing D2 RPS Connector Libraries on Microsoft Windows	Installing D2 RPS Connector Libraries on a Linux Environment

9. Restart the Documentum Server and the application server.

Installing D2 RPS Connector Libraries on Microsoft Windows

You must have administrator privileges on the local system to perform the installation.

1. On the Documentum Server:
 - a. Copy the following files to the **lib** folder depending on your Documentum Server version, as described below:
 - **D2-RPS-Connector-API.jar** from the **D2-RPS-Install-<version>.jar** extraction folder.
 - **DmcPolicyEngine.jar** from the OpenText Retention Policy Services Administrator (RPSA) webapp.
 - **DmcRecords.jar** from the OpenText RPSA webapp.
 - **DmcRps.jar** from the OpenText RPSA webapp.
 - **IDmcPolicyEngine.jar** from the OpenText RPSA webapp.
 - **IDmcRps.jar** from the OpenText RPSA webapp.
 - **IDmcRpsModules.jar** from the OpenText RPSA webapp.
 - **emc-policy-services.jar** from the OpenText Documentum Retention Policy Services WebServices.
 - **emc-retentionmarkup-services.jar** from the OpenText Documentum Retention Policy Services WebServices.

Documentum Server 7.1 and later	Documentum Server 7.0
<code><install path of Documentum> \<Java Method Server>\server \DctmServer_MethodServer \deployments\ServerApps.ear\lib</code>	<code><install path of Documentum> \<Java Method Server>\server \DctmServer_MethodServer\deploy \ServerApps.ear\lib</code>

- b. Copy **jaxb-api.jar**, **jaxb-xjc.jar**, and **jaxb-impl.jar** from **<path to D2 installation>\D2** to the **\lib** folder as described in the following table:

Documentum Server Version	Path
Documentum Server 7.0	<code><install path of Documentum> \<Java Method Server>\server \DctmServer_MethodServer\deploy \ServerApps.ear\lib\</code>

2. On the application server:
 - a. Copy **D2-RPS-Connector-API.jar** from the extraction folder to the **D2-Config\WEB-INF\lib** folder.
 - b. Copy **D2-RPS-Connector-Plugin.jar** from the extraction folder to the **D2-Config\WEB-INF\classes\plugins** folder.
 - c. Copy **D2-RPS-Connector-API.jar** and **D2-RPS-Connector-Plugin.jar** from the extraction folder to the **D2\WEB-INF\lib** folder.

3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is `<install path to web application server>\webapps\D2-Config\WEB-INF\classes`
 - b. Open for editing `D2-Config.properties`
 - c. Add and set the value of `plugin_x=` to the path for `D2-RPS-Connector-Plugin.jar`.
Use forward slashes for the file path. For example:
 - To use an absolute path: `<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/D2-RPS-Connector-Plugin.jar`
 - To use a relative path: `/plugins/D2-RPS-Connector-Plugin.jar``x` equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`
4. Return to the D2 RPS Connector installation instructions: [D2 RPS Connector Overview and Roadmap](#).

Installing D2 RPS Connector Libraries on a Linux Environment

1. On the Documentum Server:
 - a. Copy the following files to the `lib` folder depending on your Documentum Server version, as described below:
 - `D2-RPS-Connector-API.jar` from the `D2-RPS-Install-<version>.jar` extraction folder.
 - `DmcPolicyEngine.jar` from the OpenText Retention Policy Services Administrator (RPSA) webapp.
 - `DmcRecords.jar` from the OpenText RPSA webapp.
 - `DmcRps.jar` from the OpenText RPSA webapp.
 - `IDmcPolicyEngine.jar` from the OpenText RPSA webapp.
 - `IDmcRps.jar` from the OpenText RPSA webapp.
 - `IDmcRpsModules.jar` from the OpenText RPSA webapp.
 - `emc-policy-services.jar` from the OpenText Documentum Retention Policy Services WebServices.
 - `emc-retentionmarkup-services.jar` from the OpenText Documentum Retention Policy Services WebServices.

Documentum Server 7.1 and later	Documentum Server 7.0
<pre><install path of Documentum> /<Java Method Server>/server /DctmServer_MethodServer /deployments/ServerApps.ear/lib</pre>	<pre><install path of Documentum> /<Java Method Server>/server /DctmServer_MethodServer/deploy /ServerApps.ear/lib</pre>

- b. Copy `jaxb-api.jar`, `jaxb-xjc.jar`, and `jaxb-impl.jar` from `<path to D2 installation>/D2/` to the `/lib/` folder as described in the following table:

Documentum Server Version	Path
Documentum Server 7.0	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/ServerApps.ear/lib/</code>

2. On the application server:
 - a. Copy `D2-RPS-Connector-API.jar` from the extraction folder to the `D2-Config/WEB-INF/lib` folder.
 - b. Copy `D2-RPS-Connector-Plugin.jar` from the extraction folder to the `D2-Config/WEB-INF/classes/plugins` folder.
 - c. Copy `D2-RPS-Connector-API.jar` and `D2-RPS-Connector-Plugin.jar` from the extraction folder to the `D2/WEB-INF/lib` folder.
3. Activate the plug-in:
 - a. Navigate to the location of your D2 Config configuration files.
The default location is `<install path to web application server>/webapps/D2-Config/WEB-INF/classes`
 - b. Open for editing `D2-Config.properties`
 - c. Add and set the value of `plugin_x=` to the path for `D2-RPS-Connector-Plugin.jar`.
Use forward slashes for the file path. For example:
 - To use an absolute path: `<D2 Config installation path>/D2-Config/WEB-INF/classes/plugins/D2-RPS-Connector-Plugin.jar`
 - To use a relative path: `/plugins/D2-RPS-Connector-Plugin.jar``x` equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`
4. Return to the D2 RPS Connector installation instructions: [D2 RPS Connector Overview and Roadmap](#).

Installing D2-BOCS

Understanding Documentum Branch Office Caching Servers (BOCS)

D2-BOCS allows D2 to communicate with one or more Documentum Branch Office Caching Servers (BOCS) or with Documentum Accelerated Content Servers (ACS) acting in the role of a BOCS system.

BOCS servers improve file transfer performance for users by connecting to a local server even when they are remote from the Documentum Server. This allows D2 Client to use BOCS for the checking in, importing, and requesting files.

ACS are installed as part of every Documentum Server installation and allow users to bypass the application server during the transfer of files. You can use specific configurations to have D2 treat an ACS server as a BOCS server.

D2-BOCS can transfer content either synchronously or asynchronously with Documentum Messaging Systems (DMS). With asynchronous write, the upload performance over WAN is comparable to the case over LAN.

The process for transferring content to D2 using D2-BOCS is:

1. An end user checks in or imports a file.
2. D2 attempts to locate a BOCS server for the file transfer and determines if the current network of the end user is associated with a specific BOCS server network location.
3. If no BOCS server is located or responds, D2 uses a servlet on the application server as a fallback transfer mechanism.
4. If a BOCS server is located and responds, D2 establishes a connection to the BOCS URL and transfers the file.
5. If the transfer is sent:
 - Synchronously, D2-BOCS issues a **SAVE** command to indicate to BOCS that the file should be immediately saved to the Documentum Server file store.
 - Asynchronously, D2-BOCS issues a **PARK** command to indicate to BOCS that the file should be first cached on the BOCS server and then moved to the Documentum Server file store with the assistance of DMS.

Installing the D2-BOCS

Install a BOCS on a dedicated host server machine that is local to a specific network location to improve file transfer between the Documentum Server and remote end-user locations. The BOCS host machine does not need Documentum Server nor a database installed on it. Minimally, the BOCS host

machine must have Documentum BOCS and Documentum Foundation Services (DFS). The following installation contains the steps for deploying the D2-BOCS .war file on the BOCS or ACS server and then configuring other plug-ins to use the BOCS or ACS server.

1. Before installing:
 - a. Ensure Documentum BOCS is installed on a dedicated server. The *OpenText Documentum Branch Office Caching Services Release Notes* and *OpenText Documentum D2-BOCS Installation Guide* contain further instructions.
 - b. To enable asynchronous transfers, ensure Documentum Messaging Services (DMS) are installed and configured in the repository.
 - c. Ensure Documentum Foundation Classes (DFC) and D2 Client are installed on the application server.
 - d. D2-BOCS DFC client needs to be set as a Privileged Client.
 - e. `permissions.xml` from `ServerApps.ear\META-INF` has to be copied to `acs.ear\META-INF`.
2. When upgrading or installing your repository, select **Global Registry**.
3. Deploy the content of **D2-BOCS.war** on the BOCS or ACS server, as described in the following table:

On the BOCS Server	On the ACS Server
For Documentum Server 7.1: Installing D2-BOCS on a BOCS Server for Documentum Content Server 7.1	Installing D2-BOCS on an ACS Server.
For Documentum Server 7.0: Installing D2-BOCS on a BOCS Server for Documentum Content Server 7.0.	

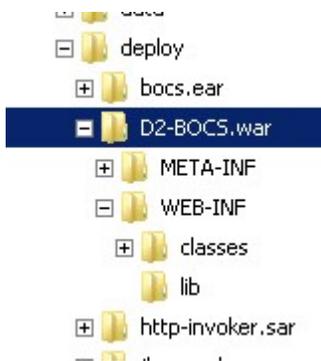
4. Configure D2 Client to enable BOCS. [Configuring D2 Client for BOCS](#) contains further instructions.
5. Set the network location identifier for the D2-BOCS servers. [BOCS and ACS Network Locations](#) contains more information and [Setting the BOCS and ACS Network Locations](#) contains instructions for setting the network location identifier parameter.
6. Configure other plug-ins to interact with D2-BOCS as described in the following table:

Plugin	Instructions
O2	Configuring O2 for BOCS
C2	Configuring C2 for BOCS

7. You can configure the following optional settings:
 - [Enabling BOCS Content Transfer with Non-Anonymous Certificate Based SSL](#) contains further instructions on enabling content transfer using BOCS with non-anonymous certificate-based SSL connections.
 - [Enabling Compressing for Upload and Download](#) contains further information and instructions on enabling file compression for upload and download through D2-BOCS.
 - [Enabling Asynchronous BOCS Write](#) contains further instructions for enabling asynchronous write.
8. Verify the installation of D2-BOCS and the status of the BOCS or ACS server. [Checking D2-BOCS Installation](#) contains further instructions.

Installing D2–BOCS on a BOCS Server for Documentum Server 7.0

1. Download and extract the contents of `D2-BOCS.war` to `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/`. Ensure the extracted folder is named `D2-BOCS.war`, as shown in the following screenshot:



2. Delete the `D2-BOCS.war` file.
3. Copy all `.jar` files from `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/bocs.ear/lib/` to `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/` except for the following files:
 - `commons-collections_<version>.jar`
 - `commons-io-<version>.jar`
 - `commons-lang-<version>.jar`
 - `spring-context-support-<version>.release.jar`
4. Delete the following files from `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/`
 - `jsr<version>_api.jar`
 - `jaxb-api.jar`
 - `stax-api-<version>.jar`

5. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Installing D2-BOCS on a BOCS Server for Documentum Server 7.1 or later

1. Download and extract the contents of `D2-BOCS.war` to `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/`. Ensure the extracted folder is named `D2-BOCS.war`, as shown in the following screenshot:



2. Delete the `D2-BOCS.war` file.
3. Copy all `.jar` files from `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/bocs.ear/lib/` to `<installation path of Documentum>/<server version>/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/` except for the following files:
- `commons-collections_<version>.jar`
 - `commons-io-<version>.jar`
 - `commons-lang-<version>.jar`
 - `spring-context-support-<version>.release.jar`
 - `cxfrt-api-<version>.jar`
 - `cxfrt-bindings-soap-<version>.jar`
 - `cxfrt-bindings-xml-<version>.jar`
 - `cxfrt-core-<version>.jar`
 - `cxfrt-databinding-jaxb-<version>.jar`
 - `cxfrt-features-clustering-<version>.jar`
 - `cxfrt-frontend-jaxws-<version>.jar`
 - `cxfrt-frontend-simple-<version>.jar`
 - `cxfrt-transport-http-<version>.jar`
 - `cxfrt-ws-addr-<version>.jar`
 - `cxfrt-ws-policy-<version>.jar`

4. Delete the following files from `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/`
 - `jsr<version>_api.jar`
 - `jaxb-api.jar`
 - `stax-api-<version>.jar`
5. Create a dummy file named `D2-BOCS.war.dodeploy` in `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/`
6. Copy `anonymous-service-handler-chain.xml` from the `configs.jar` folder to the `acs/ws/ws/` folder as described in the following table:

Folders	Paths
Original location	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/bocs.ear/lib/configs.jar/anonymous-service-handler-chain.xml</code>
Destination	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/bocs.ear/D2-BOCS.war/WEB-INF/lib/bocs-ws.jar/com/documentum/acs/ws/ws/</code>

7. Copy `authorized-service-handler-chain.xml` from the `configs.jar` folder to the `services/ws/` folder as described in the following table:

Folders	Paths
Original location	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/bocs.ear/lib/configs.jar/authorized-service-handler-chain.xml</code>
Destination	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/bocs.ear/D2-BOCS.war/WEB-INF/lib/D2FS-Generated-<version>.jar/com/emc/d2fs/dctm/api/services/ws/</code>

8. Navigate to `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/classes`, open `dfc.properties` in a text editor.
 - a. Add the following line:

```
dfc.bof.classloader.enable_extension_loader_first=false
```
 - b. Add the following lines for docbroker/globalregistry settings:

```
dfc.docbroker.host[0]=
dfc.docbroker.port[0]=
dfc.globalregistry.repository=
dfc.globalregistry.username=
dfc.globalregistry.password=
```
9. Create `jboss-deployment-structure.xml` and add the following lines:

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure>
  <deployment>
```

```

    <!-- Exclusions allow you to prevent the server from automatically
         adding some dephenencies -->
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="org.hibernate" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>

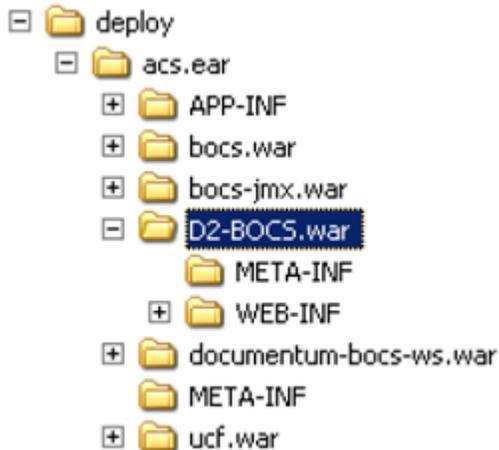
```

10. Restart the BOCS.
11. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Installing D2–BOCS on an ACS server

By default, D2 does not recognize an ACS server as a BOCS system. Perform the following steps if you want to include an ACS server as a BOCS system.

1. Download and extract the contents of **D2-BOCS.war** to `<install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear`. Ensure the extracted folder is named **D2-BOCS.war**, as shown in the following screenshot:



2. Delete the **D2-BOCS.war** file.
3. If you are using Documentum Server version 7.1:
 - a. Copy **authorized-service-handler-chain.xml** from the **configs.jar** folder to the **services/ws/** folder as described in the following table:

Folders	Paths
Original location	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/lib/configs.jar/authorized-service-handler-chain.xml</code>
Destination	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/D2FS-Generated-4.2.0.jar/com/emc/d2fs/dctm/api/services/ws/</code>

- b. Navigate to `<install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/classes/`, open `dfc.properties` in a text editor, and add the following line:

```
dfc.bof.classloader.enable_extension_loader_first=false
```

Add the following lines for docbroker/globalregistry settings:

```
dfc.docbroker.host[0]=
```

```
dfc.docbroker.port[0]=
```

```
dfc.globalregistry.repository=
```

```
dfc.globalregistry.username=
```

```
dfc.globalregistry.password=
```

4. Enable the D2-BOCS module:

- a. Open `application.xml` in a text editor from the location described in the following table:

Documentum Server Version	Path
Documentum Server 7.1	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/META-INF/</code>
Documentum Server 7.0	<code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear/META-INF/</code>

- b. Add the following lines:

```
<module id="D2-BOCS"><web><web-uri> D2-BOCS.war</web-uri><context-root>/D2-BOCS</context-root></web></module>
```

5. Navigate to `~<DOCUMENTUM_HOME>\<Java Method Server>\server\DctmServer_MethodServer\deployments\acs.ear\META-INF` and Open the `jboss-deployment-structure.xml`. Create a sub-deployment section for `D2-BOCS.war` and add the exclusions:

```
<sub-deployment name="D2-BOCS.war">
  <exclusions>
    <module name="org.apache.log4j"/>
    <module name="org.slf4j"/>
    <module name="org.slf4j.impl"/>
    <module name="org.hibernate"/>
  </exclusions>
</sub-deployment>
```

6. Restart the JMS.
7. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Configuring D2 Client for BOCS

1. Navigate to `D2/WEB-INF/classes/` and open `D2FS.properties`
2. Uncomment the line: `#pluginsOrder=D2-BOCS,C2,O2`
3. Uncomment and set the line: `D2-BOCS=true`

4. If D2-BOCS is deployed on the ACS server running on the Documentum Server, add the line **includeAcsServer=true**

If you do not want to use an ACS server for BOCS purposes, set the value to **false**

If the value is set to **true** but an ACS server is not configured for BOCS, the communication fails and D2 uses the application server for file transfer.

5. Configure the following parameters:

Parameter	Description
D2-BOCS	Set to true to enable Branch Office Caching Services (BOCS) in D2 Client if D2-BOCS is deployed on one or more BOCS servers.
includeAcsServer	Set to true to enable BOCS if D2-BOCS is deployed on the Accelerated Content Services (ACS) server on the Documentum Server.
minFileSizeForBocs	Set a minimal size in bytes for determining whether to use D2-BOCS for file download and upload. Depending on the includeAcsServer parameter, if the file size is smaller than the minFileSizeForBocs , D2 uses the ACS or a direct D2 Client download.
cacheBocsUrl	Set to true to force the Documentum Foundation Classes cache location to load before running any download or upload attempt. This requests a load on startup configuration. By default, this parameter is set to false .

6. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Configuring O2 for BOCS

If you have O2 version 2.1.0 or later, you can configure O2 to use BOCS for file transfer by performing the following steps.

1. Add and set the value of **plugin_x=** to the path for **O2-Plugin.jar** in **D2-BOCS.properties**. The location of **D2-BOCS.properties** can differ based on the server type. The following table describes the locations:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/Classes/</code></p>
ACS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/</code></p>

Use forward slashes for the file path. For example:

- To use an absolute path: `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/classes/plugins/O2-Plugin.jar`
- To use a relative path: `/lib/plugins/O2-Plugin.jar`

x equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`

2. Copy `O2-API.jar` from the O2 plug-in download or from the install path of your O2 installation to the `lib` folder of each BOCS server. The following table describes the locations of the `lib` folders:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/</p>
ACS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/</p>

3. Copy **O2-plugin.jar** from the O2 plug-in download or from the install path of your O2 installation to the **plugins** folder of each BOCS server. The following table describes the locations of the **plugins** folders:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/</code></p>
ACS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/</code></p>

- Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Configuring C2 for BOCS

If you have C2 version 2.1.0 or later, you can configure C2 to use BOCS for file transfer by performing the following steps.

- Add and set the value of `plugin_x=` to the path for `C2-Plugin.jar` in `D2-BOCS.properties`. The location of `D2-BOCS.properties` can differ based on the server type. The following table describes the locations:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/Classes/</code></p>
ACS	<p>For Documentum Server 7.1, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/</code></p> <p>For Documentum Server 7.0 and older, use <code><install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/</code></p>

Use forward slashes for the file path. For example:

- To use an absolute path: `<install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/classes/plugins/C2-Plugin.jar`
- To use a relative path: `/lib/plugins/C2-Plugin.jar`

x equals the number of previous plug-in plus one. If no other plug-in is installed, use `plugin_1`

2. Copy `C2-API.jar` from the C2 plug-in download or from the install path of your C2 installation to the `lib` folder of each BOCS server. The following table describes the locations of the `lib` folders:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/</p>
ACS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/</p>

3. Copy **C2-plugin.jar** from the C2 plug-in download or from the install path of your C2 installation to the **plugins** folder of each BOCS server. The following table describes the locations of the **plugins** folders:

Server type	Location
BOCS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/</p>
ACS	<p>For Documentum Server 7.1, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/</p> <p>For Documentum Server 7.0 and older, use <install path of Documentum>/<Java Method Server>/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/</p>

- Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Checking D2-BOCS Installation

You can perform the following steps to confirm the state of your BOCS installation.

- To check for the correct installation of D2-BOCS on a BOCS server, navigate to **http://<bocs_server_name>:8086/D2-BOCS/** in Microsoft Internet Explorer.

If D2-BOCS is installed and running correctly, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
  -<bocs version="4.x.x build xxx" server_time="x.xxxx">
    <plugins />
  </bocs>
```

If you have plug-ins installed, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
  -<bocs version="4.x.x build xxx" server_time="x.xxxx">
    <plugins>C2 v4.x.x build xx</plugins>
  </bocs>
```

- To check for the correct installation of D2-BOCS on an ACS server, navigate to **http://<content_server_name>:9080/D2-BOCS/** in Microsoft Internet Explorer.

If D2-BOCS is installed and running correctly, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
  -<bocs version="4.x.x build xxx" server_time="x.xxxx">
    <plugins />
```

```
</bocs>
```

If you have plug-ins installed, the browser shows the following message:

```
<?xml version="1.0" encoding="utf-8">
-<bocs version="4.x.x build xxx" server_time="x.xxxx">
  <plugins>C2 v4.x.x build xx</plugins>
</bocs>
```

3. To check the status of your BOCS server, navigate to **http://<bocs_server_name>:8086/bocs/servlet/ACS**
If the BOCS server is running, the browser shows the answer message **ACS Server is running**
You can also check for the status of the BOCS server through D2 Client and D2 Config.
4. To check the status of your ACS server, navigate to **http://<content_server_name>:9080/ACS/servlet/ACS**
If the BOCS server is running, the browser shows the answer message **ACS Server is running**
You can also check for the status of the BOCS server through D2 Client and D2 Config.
5. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Enabling BOCS Content Transfer with Non-Anonymous Certificate Based SSL

1. Enable non-anonymous certificate-based Secure Sockets Layer (SSL) on your Documentum Server. The *OpenText Documentum Server Administration and Configuration Guide* contains further information and instructions on enabling non-anonymous certificate based SSL.
2. Enable SSL on the BOCS and DMS Servers. The *OpenText Documentum Server Distributed Configuration Guide* contains instructions for enabling SSL.
3. Copy **dfc.keystore** from the Documentum Server to the BOCS server.
For example, **<install path to Documentum>/dba/secure/**
4. Navigate to **<install path of web application server>/webapps/D2-BOCS/WEB-INF/classes** and open for editing **dfc.properties**
5. Add the following lines:
dfc.security.ssl.truststore=<path to dfc.keystore>
dfc.security.ssl.truststore_password=<password>
6. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Enabling Compression for Upload and Download

1. To enable compression between the D2 Client application and D2-BOCS:
 - a. Navigate to and open **D2-BOCS.war/WEB-INF/web.xml**
 - b. Uncomment the lines:

```
<filter>
  <filter-name>CompressingFilter</filter-name>
  <filter-class>
    com.planetj.servlet.filter.compression.CompressingFilter
  </filter-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>>true</param-value>
  </init-param>
  <init-param>
    <param-name>compressionThreshold</param-name>
    <param-value>1024</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>CompressingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

2. To enable compression between D2-BOCS and Documentum Server for non-C2/O2 deployments:
 - a. Navigate to and open `/D2/WEB-INF/classes/settings.properties`
 - b. Add or set the following parameters:

```
applet.download.compression.enabled = true
applet.upload.compression.enabled = true
applet.upload.compression.threshold = 1024
applet.upload.compression.extensions =
doc, docx, xls, xlsx, ppt, pptx, pdf, txt
```
 - c. Navigate to and open `/D2-BOCS/WEB-INF/classes/D2-BOCS.properties`
 - d. Uncomment and set the following parameter:

```
compressedExtensions = doc, docx, xls, xlsx, ppt, pptx, pdf, txt
```
3. Enable D2-BOCS:
 - a. Navigate to and open `/D2/WEB-INF/classes/D2FS.properties`
 - b. Add or set the following parameters:

```
D2-BOCS=true
includeAcsServer=true
```
4. Return to the BOCS installation instructions: [BOCS Pre-requisite and Roadmap](#).

Enabling Asynchronous BOCS Write

1. Log in to Documentum Administrator.
2. Navigate to [Global Repository] > Administration > Distributed Content Configuration > Distributed Transfer.
3. Open the properties of the `ContTransferConfig` object.

4. Set the **ACS Write** property to **Synchronous and Asynchronous Write**.
5. Save the object properties.

BOCS and ACS Network Locations

You must set the **networkLocationId** for each D2-BOCS server based on the **Network Location Identifier** property as specified in Documentum Administrator. Setting these values ensures that the D2-BOCS server communicates with adjacent BOCS or ACS servers rather than a remote BOCS or ACS server. If there are several network locations associated with a BOCS or ACS server configuration, you should choose the one that is referenced only by that BOCS or ACS server.

For example, if you have D2-BOCS deployed to:

- A BOCS server (B1) whose network locations have the identifiers NL1 and NL2.
- A BOCS server (B2) whose network locations have the identifiers NL2 and NL3.

For the D2-BOCS server on the B1 server, set **networkLocationId=NL1**, and for the D2-BOCS server on the B2 server, set **networkLocationId=NL3**.

ACS servers are not typically associated with a network location. In order for failover to D2-BOCS on an ACS server to work correctly, you must associate a network location with the ACS server. You can do this by creating a Network Location in Documentum Administrator and setting the **networkLocationId** for the ACS server.

For example, if you have D2-BOCS deployed to an ACS server (A1) whose network location has the identifier NL0, for the D2-BOCS server on the A1 server, set **networkLocationId=NL0**.

Setting the BOCS and ACS Network Locations

1. Open **D2-BOCS.properties** in a text editor.

The location of **D2-BOCS.properties** can differ based on the server type. The following table describes the locations:

Server type	Location
BOCS	<server location>/DctmServer_BOCS/ deploy/D2-BOCS .war/WEBINF/Classes/
ACS	<server location>/DctmServer_BOCS/ deploy/acs.ear /D2-BOCS.war/WEBINF/Classes/

2. Set the network location for the D2-BOCS server on each BOCS and ACS server you are using.

Using D2-BOCS for Download

Download performance as tested on a 300ms latency 2Mbps bandwidth connection is compared in the following table:

Download in WAN (seconds)	Application Server	ACS	Downloading for the first time through BOCS	Downloading a second time through BOCS
10kb	1.3	1.3	1.3	<0.1
100kb (compression ratio 50%)	2.2	1.9	1.9	<0.1
1MB (compression ratio 50%)	3.8	3.6	3.5	0.2
10MB (compression ratio 50%)	24.5	23.8	23.7	0.8
100MB (compression ratio 50%)	225.9	219.9	221.4	6.3

The tests show that:

- There are no significant differences if you use an application server or use an ACS the first time content is downloaded.
- BOCS significantly improves performance from the second time content is downloaded onward.

To use D2-BOCS:

1. Navigate to and open `<install path to web application server>/webapps/D2/WEB-INF/classes/D2FS.properties`
2. Tune the `minFileSizeForBocs` parameter to force D2 to not use BOCS for small files when performing both download or upload operations. Use the breakeven file size where using D2-BOCS does not have a positive or negative impact on file transfer performance.

Conduct tests in the production environment because the optimal value depends on the network conditions. The OpenText internal test results are compared in the following table:

Download in WAN (seconds)	Application Server	ACS	Downloading for the first time through D2-BOCS	Downloading a second time through D2-BOCS
10kb	1.3	1.3	12.0	10.0
100kb (compression ratio 50%)	2.2	1.9	12.8	10.1
1MB (compression ratio 50%)	3.8	3.6	15.2	10.6

Download in WAN (seconds)	Application Server	ACS	Downloading for the first time through D2-BOCS	Downloading a second time through D2-BOCS
10MB (compression ratio 50%)	24.5	23.8	36.2	13.3
100MB (compression ratio 50%)	225.9	219.9	258.5	25.4

The comparison shows that D2-BOCS:

- Significantly improves performance on subsequent downloads when downloading a roughly 5MB size file at a compression ratio of 50%.
 - Introduces an overhead of 10 seconds with the first and the subsequent download as a function of WAN conditions.
3. In order to enable the D2-BOCS server to load various caches when it starts up, open its **D2-BOCS.properties** file and update the **LoadOnStartup** parameter by setting its value to a comma-separated list of repository names for which caches should be loaded at startup time. For example, `LoadOnStartup=repo1,repo2`.

Note that the corresponding username and password properties for each listed repository need to be set in the D2 global registry keystore. For example,

```
LoadOnStartup.repo1.username=dadmin1
LoadOnStartup.repo1.password=password1
LoadOnStartup.repo2.username=dadmin2
LoadOnStartup.repo2.password=password2
```

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

```
LoadOnStartup.*.username=dadmin
LoadOnStartup.*.password=password
```

4. To initialize an expensive precache process of BOCS locations on D2 startup instead of the initial content transfer:
- Set the following parameters:
 - LoadOnStartup.docbase**
 - LoadOnStartup.username**
 - LoadOnStartup.password**
 - LoadOnStartup.domain**
 - Set the **cacheBocsUrl** parameter to **true** in the following two locations:
 - `<install path to web application server>/webapps/D2/WEB-INF/classes/D2FS.properties`
 - `<install path to D2-BOCS>/WEB-INF/classes/D2-BOCS.properties`

Using D2-BOCS for Upload

Do not use D2-BOCS for upload unless you upload large content. The OpenText internal test results for upload performance are compared in the following table:

Upload in WAN (seconds)	Application Server	ACS	Synchronous write through D2-BOCS	Asynchronous write through D2-BOCS
10kb	2.2	2.0	25.2	23.1
100kb (compression ratio 50%)	3.7	3.3	25.8	24.5
1MB (compression ratio 50%)	6.9	6.1	28.4	24.6
10MB (compression ratio 50%)	26.4	25.5	51.8	28.1
100MB (compression ratio 50%)	230.3	230.1	267.8	38.4

The comparison shows that:

- With small content, both synchronous and asynchronous write introduce an overhead of about 20 seconds as a function of WAN conditions.
- With large content (>1MB), asynchronous write has superior performance to synchronous write.
- With larger content (>10M), asynchronous write performs better than using ACS.

Uninstalling D2

Uninstalling D2

1. Use Documentum Administrator or DQL/API to look for and remove the following artifacts:
 - All object types added by the D2 DAR, which are prefixed with **D2** and **D2_**
 - All methods added by the D2 DAR, which are prefixed with **D2** and **D2_**
 - All jobs added by the D2 DAR, which are prefixed with **D2** and **D2_**
 - Run the following DQL queries to make sure nothing exists:
 - `select * from dm_type where name like '%d2%'`
 - delete x3_preferences objects

Note: If you have installed the plugins, search for types related to c2 and o2.

 - Once you removed all the mentioned artifacts, run the consistency checker job and check for errors.
2. On the Documentum Server, refer to the installation log file to delete the .jar files that were copied to the Java Method Server.
3. On the web application server:
 - a. Disable or uninstall the D2 and D2 Config web applications if your web application server provides a user interface for removing the web application.
 - b. Delete the folders that contained the D2 and D2 Config files.

Troubleshooting the Installation

D2 Client and D2-Config deployment fails on Jboss server

Unable to launch D2 on Tomcat 8.0.5. Error while fetching repositories

Problem

When logging into D2 an “error while fetching repositories” message is displayed on Linux 6.5 with Tomcat 8.0.5, Windows 2008 with Tomcat 8.0.5, and Windows 2012 R2 with Tomcat 8.0.x.

Cause

Requires additional configuration.

Resolution

Running Tomcat 8.0.5 with D7.1SP1 and DFS7.1.1 requires modifications to `catalina.properties`: `jnlpc.com.rsa.cryptoj.fips140loader=true` File location: `file <CATALINA_HOME>/conf/catalina.properties`

Running Tomcat 8.0.x with D7.1 CS and DFS7.1 requires copying `cryptojce.jar`, `cryptojcommon.jar`, and `jcmFIPS.jar` files to the `/D2/WEB-INF/lib` and `/D2-Config/WEB-INF/lib` folders. Also, modify the `catalina.properties` file as mentioned above.

Removing old versions of ctx.cab from D2-Config machines

Problem

Occasionally, old versions of the files in `ctx.cab`, an Active-X add-on used on the machines running D2-Config are present and need to be manually deleted and re-installed.

Resolution

Locate the `ctx` add-on on the client machine where D2-Config is accessed and delete them. The add-on can usually be found at `C:\Windows\Downloaded Program Files`. If they are not visible

in Windows Explorer, delete them through command prompt. After deletion, access D2–Config on the machine again to get a new version of ctx.cab.

Disabling xQuery when Using the xPlore Search Engine

By default, the Documentum Foundation Classes search service generates results in xQuery format. In some situations when the xPlore indices are not current or xPlore is not performing as expected, the results from xQuery may not match repository data. If that is the case and repairing the xPlore indices or improving xPlore performance is not possible, then the following change may be made to disable xQuery when using the xPlore search engine.

1. To disable xQuery for all applications that use a shared **dfc.properties**, navigate to and open the shared **dfc.properties** file, for example at **%DOCUMENTUM%/config/dfc.properties**
If you want to disable xQuery for specific applications such as D2, navigate to and open the application–specific **dfc.properties** instead of the shared properties file, for example at **%APPSERVER%/webapps/D2/WEB-INF/classes/dfc.properties**
2. To disable xQuery, find or add the entry **dfc.search.xquery.generation.enable** and set the value to **false**

Unable to Access D2 Using Microsoft Internet Explorer

Problem

D2 cannot run ActiveX controls on Microsoft Internet Explorer.

Cause

Microsoft Internet Explorer blocks D2 URLs from running ActiveX controls and MSXML.

Resolution

Make D2 URLs a part of the intranet or Trusted Security Zone to allow D2 ActiveX controls and MSXML.

Configuring Microsoft SQL Server 2008

Problem

Error message received when saving D2 options or a template:

```
<date and time> [ERROR] - com.emc.d2.web.servlets.D2Context : {}
com.documentum.fc.client.DfServiceException:

at com.documentum.fc.client.transaction.impl.Transaction.commit
(Transaction.java:66) [dfc.jar:na]
```

```
at com.documentum.fc.client.transaction.impl.TransactionManager.commit
(TransactionManager.java:30) [dfc.jar:na]
```

```
at com.documentum.fc.client.impl.session.SessionManager.commitTransaction
(SessionManager.java:351) [dfc.jar:na]
```

Cause

Microsoft SQL Server 2008 requires an additional setting on the Documentum Server machine.

Resolution

Use the `SQL ALTER` command to set `READ_COMMITTED_SNAPSHOT` to the `ON` state.

Unable to Open a New Window or New Connection. No Resource Available.

Problem

Warning dialog box that says `Unable to open a new window or new connection. No resource available.`

Cause

There is a problem with the browser cookie settings.

Resolution

1. Open an instance of Microsoft Internet Explorer.
2. Click **Tools > Internet Options**.
3. On the **General** tab, click **Delete Cookies**.
4. Click **OK**.

DfRegistryWin32.DLL is Already Loaded in Another Classloader

Problem

When D2 Config and D2 Client (any version) are both running, only one web application can perform Documentum operations such as reading, editing, and checking in, and checking out. Attempting to perform Documentum operations on the other web application leads to the error message: `DfRegistryWin32.dll is already loaded in another Classloader.`

Cause

The error is shown when both applications use the same IBM JVM with no cluster configuration in an IBM WebSphere 6.1 environment on a Microsoft Windows installation.

Resolution

Follow the documentation for Documentum Administrator installation via the Web Development Kit or Webtop on IBM Websphere 6.1, or set the registry mode in the `dfc.properties` file from `windows` to `file`:

```
dfc.registry.mode = file
```

```
dfc.registry.file = ${dfc.data.user_dir}/documentum.ini
```

IBM AIX and Apache Tomcat 6.0 Crashing the Server

Problem

When using Apache Tomcat with an IBM AIX server, you can encounter issues with the number of simultaneously opened files. A number of file channels reading `D2-Web.jar` are opened for each user session, which exceeds the operating system limit and crashes the application server. The problem does not occur in Sun Solaris or Microsoft Windows environments.

Cause

The web application must directly access class files instead of the archive.

Resolution

Use the validated workaround for this problem:

1. Stop your application server.
2. Remove `D2-Web.jar` from the `WEB-INF/lib/` folders for D2 Config and D2 Client.
3. Extract the contents of `D2-Web.jar` to the `WEB-INF/classes/` folders for D2 Config and D2 Client.

Rename the file to `D2-Web.zip` if your unzip tool does not recognize the JAR file.

Files Corrupting During Export

Problem

When exporting a file from the repository to your local file system using D2 Client (any version), the file is corrupted. This issue exists in all compatible web servers except Tomcat 5.5.

Cause

While using the `Save As` dialog box, the session times out, and the file is corrupted.

Resolution

Configure the HTTP 1.1 connector `connectionTimeout` global setting for your web application server to wait longer before disconnecting the session.

While the parameter defaults to 60 seconds when not set, installation of the web server sets the parameter to 20 seconds. The documentation for your web server contains the default value and further instructions.

For example, in Tomcat 6.x:

1. Navigate to `<Tomcat installation path>/conf/` and open `server.xml`.
2. Locate the line `<Connector port="port" protocol="HTTP/1.1" connectionTimeout="timeout duration" />`.
3. Change `timeout duration` to the duration you want in milliseconds, such as 60000.

D2 Caching and File-Cleaning Services Fail to Operate

Problem

D2 caching services and temporary D2 file-cleaning services fail to operate normally due to file deadlock.

Cause

If D2 is deployed on multiple JVMs on the same application server or machine, the JVMs by default share the same folder and lock files from each other.

On a Linux environment, the error is caused frequently by JVMs being run by different users.

On Microsoft Windows systems, the error is caused by critical files being overwritten.

Resolution

Set up private Java temporary directories for each JVM instance.

To define a specific Java temporary directory, add the parameter `-Djava.io.tmpdir=/tmp/my_jvm_tmpdir` to the JVM launch command line.

Null Pointer Exception When Using Reverse-proxy IIS 7 to Import a File Larger Than 25 MB

Problem

Import fails when using IIS 7 as a reverse proxy for D2 and importing a file larger than 25 MB. The Java Console log of the browser machine displays `Null Pointer Exception`.

Cause

IIS is not configured to support large files.

Resolution

1. Log in to IIS Manager.
2. Click **Default Website**.
3. Navigate to **IIS > Request Filtering**.
4. In the view that opens, select **File Name Extensions**.
5. Right-click the view and select **Edit Feature Settings** in the context menu.
6. In **Maximum allowed content length**, select a larger value.

Content Transfer Does Not Go Through the BOCS Server

Problem

Content transfer does not go through the BOCS server.

Cause

If the ACS server is not running, the BOCS server is not called and D2 Client uses the application server servlet.

Resolution

To identify the issue, review the D2 Client log, located by default in **C:/logs/D2.log** for the following line:

```
[DFC_ACS_NO_ACS_FOR_DOCBASE] Cannot find ACS servers for docbaseId=xxx  
docbaseName=xxxxxxx from docbrokers
```

If the line appears and the ACS server is running, wait roughly 5 minutes for D2 Client to re-try communication with the ACS server and then reconnect.

Running a Java Console and logging at level 5 may indicate why the connection is failing.

Slow File Transfer When Using a Linux-based Operating System

Problem

D2 file transfer is slower than Documentum Administrator when run on a Linux-based operating system.

Cause

Known issue with random number generation on Linux-based operating systems.

Resolution

1. Manually start the random generator daemon by typing `/sbin/rngd -b -r /dev/urandom -o /dev/random` as the root user.

You can also include `-Djava.security.egd=file:///dev/./urandom` in your web application server startup script. Refer to your web application server documentation for further information.

2. If your Documentum Server is Linux-based, you may need to modify the Java Method Server to use the random number generator. Check the number of `entropy_avail` events by typing `cat /proc/sys/kernel/random/entropy_avail`

If the `entropy_avail` did not increase, navigate to and open `startMethodServer.sh`, then add the line `Djava.security.egd=file:///dev/urandom`

For Java 5 or later, use the line `-Djava.security.egd=file:///dev/./urandom`

java.lang.ClassNotFoundException when Running an Installer Package Problem

Attempting to run an installer .jar file fails, and the following message is shown:

```
java <D2 Installer>.jar
Exception in thread "main" java.lang.NoClassDefFoundError: <D2 Installer>
Caused by: java.lang.ClassNotFoundException: <D2 Installer>.jar
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
Could not find the main class: <D2 Installer>.jar. Program will exit
```

Cause

There is a problem with the Java Runtime Environment classpath.

Resolution

Manually launch the installer .jar file:

1. Open a command prompt, such as xterm.
2. Navigate to the folder containing the D2 Installer .jar file.
3. Type `java -jar <installer filename>.jar`

No JMS Server Available Exception When Trying Java Method Server (JMS) Failover Problem

On an Documentum Server cluster environment, D2 returns a **No JMS Server Available** exception when trying to perform an import operation. The JMS log shows an authentication error.

Cause

Authentication error for HA-JMS (High Availability JMS).

Resolution

1. Log in to Documentum Administrator.
2. Navigate to **Client Rights Management > Privileged Clients**.
3. Under **Manage Clients**, select **Enable trusted login** and **Enable trusted server privilege** for the Documentum Server Documentum Foundation Classes.

InstallException When Installing the Same D2 DAR Files

Problem

Installation of the same DAR files for the second time throwing error as: "InstallException: Unexpected error installing dar.."

Cause

This happens when an existing project **D2-DAR** found in the workspace that is project **D2-DAR** already has core project **DocumentumCoreProject** in its classpath.

Resolution

Use the validated workaround for this problem:

1. Delete all the folders in `\Documentum\product\7.0\install\composer\ComposerHeadless\darinstallerworkspaces`.
2. Close the opened **dardeployer**.
3. Launch the **dardeployer** and try to install the same DAR again.

Configuration Files

Configuration File Locations

Navigate to `<install path of web application server>/webapps/D2-Config/WEB-INF/classes` for the D2 Config configuration files:

- `D2-Config.properties`
- `dfc.properties`
- `logback.xml`

Navigate to `<install path of web application server>/webapps/D2/WEB-INF/classes` for the D2 Client and D2FS configuration files:

- `applicationContext.xml`
- `settings.properties`
- `logback.xml`
- `D2FS.properties`
- `dfc.properties`
- `shiro.ini`

If you did not rename `shiro-base.ini` during configuration of authentication, `shiro.ini` may not exist.