EMC® Documentum® D2

Version 4.2

Installation Guide

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Legal Notice

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com. Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries. All other trademarks used herein are the property of their respective owners.

Documentation Feedback

Your opinion matters. We want to hear from you regarding our product documentation. If you have feedback about how we can make our documentation better or easier to use, please send us your feedback directly at ECD.Documentation.Feedback@emc.com.

Table of Contents

Preface

EMC® 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.

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
June 2016	Added additional information about the `<optional Lockbox passphrase>` to the Configuring Lockbox on the Documentum Content Server, page 23 section. Added information about the `processXploreResultSet` parameter to the Configuring D2 Client, page 58 section.
June 2015	Added a new section Unable to Receive the Task after Using Update Performer Action, page 133.
December 2014	Added the information related to `docbroker/globalregistry` settings for `dfc.properties` file in Installing D2-BOCS chapter.
November 2014	• Removed the information related to `dm_event_sender_java` method from the Configuring the D2EventSenderMailMethod, page 70 topic. • Updated the relative file path for configuring O2 and C2 plug-in for BOCS.
September 2014	Added the following topics in Chapter 7, Configuring Authentication: • Single Sign On (SSO) Authentication for D2, page 75 • Configuring TrustedReverseProxy, page 78 • Adding the Administrator Login Name to the d2fs-trust.properties file, page 79 • Adding the Administrator Password to the D2 Lockbox, page 79 • Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server, page 79

Revision Date	Description
June 2014	Added a step for removing serializer-x.x.x.jar file from the JMS libraries in the Upgrading D2, page 10 topic.
May 2014	• Added a note about password restriction for D2Method passphrase parameter in the Linux environment • Added a note about running D2 and Webtop in the same or different application servers while working with Kerberos SSO. • Updated information about jar files in Step 4b and 5b in the Installing D2 RPS Connector Libraries on Microsoft Windows, page 96 and Installing D2 RPS Connector Libraries on a Linux Environment, page 98 sections respectively.
April 2014	• Added Step 4 about setting up the lockbox file for Documentum Process Engine in the Configuring Lockbox on the Documentum Content Server, page 23 section. • Updated Step 7 and Step 12 in the Installing D2 on IBM WebSphere, page 39 section. • Added Step 1(d) in the Configuring Lockbox on the Documentum Content Server, page 23 section.
March 2014	• Updated the folder path in the Installing D2 on Redhat JBOSS, page 42 section. • Added Step 5 and Step 6 in the Configuring Microsoft Windows NT Unified Logon (NTLM), page 75 section. • Updated the Configuring Kerberos, page 76 section and added a reference to the Single Sign On section. • Updated the Content Server versions and path in the Configuring Lockbox on the Documentum Content Server, page 23 section in context of copying the D2.lockbox file.
January 2014	Initial publication.

Chapter 1

Getting Started

This chapter contains the following topics:

Know Before You Start

Before installing, make sure you know:

- How to set CLASSPATH env.
- The install paths for Documentum, Content Server, Java Method Server (JMS) on both machines, 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 *EMC Documentum D2 Release Notes* for your corresponding version for compatibility matrices and updated installation procedures.
2. Make sure you have installed:
   - Documentum Content 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.
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	Documentum Foundation Services (DFS) Software Development Kit (SDK) Download the DFS SDK zip file for the version of Documentum Content Server you are using. To locate the file, use the search feature for DFS SDK *x* SP*x* For example, if you use Documentum Content Server 6.7, search for DFS SDK 6.7. If you use Documentum Content Server 6.7 SP1, search for DFS SDK 6.7 SP1	All content within this archive are needed for installation.
D2_*x*.zip	Contains the D2 core installer.	D2-Installer-<*version*>.jar.
HR Config D2 4.*x* – Export-Config.zip	Contains the sample application to import after installing D2. The application returns an error when certain configurations do not exist. Download and import the sample configuration or a previous configuration to ensure that required configurations exist for D2 to run without errors. If you are importing a separate application configuration, you do not need to import the provided sample.	All content within this archive are needed for installation.

6. If you are upgrading D2, follow the instructions in Upgrading D2, page 10.
7. Follow the instructions for installing D2 in Instructions for Installing D2, page 11.

Upgrading D2

1. Delete existing user preferences with the DQL query `delete d2c_preferences objects;`
2. Stop Content Server, JMS services, and web application services.
3. Delete the cookies and cache of the web browsers.
4. Back up configuration files. Locations of Configuration Files, page 135 contains the locations of configuration files.
5. Remove all poi-<*type*>-3.6.jar files from the JMS libraries located as described in the following table. If you do not remove the 3.6 files, the installer does not replace the files and D2 may not function correctly.

Content Server 7.1 and later	Content Server 6.7 SPI to 7.0	Content Server 6.6 and 6.7
<install path of Documentum>/*<JBOSS version>*/server/Dctm-Server_Method-Server/deployments/Server-Apps.ear/lib	*<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/lib	*<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/APP-INF/lib

6. Remove `serializer-x.x.x.jar` file from the JMS libraries located at `bpm.ear/lib`.
7. 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.
8. Follow the Instructions for Installing D2, page 11.

Instructions for Installing D2

The following installation contains the steps for installing the D2-API library files on the Documentum Content 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 Content 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.
   b. Stop the Documentum Content Server repository and JMS services.
   c. Follow the instructions for installing the D2 API libraries and extracting `D2-DAR.dar`, `D2Widget-DAR.dar`, and `Collaboration_Services.dar`, with or without compatibility with Documentum Process Engine, as described in the following table. D2-API is a set of libraries for the Documentum Content Server and the JMS enabling D2 Methods to be run on the Content Server.

Operating system of your Documentum Content Server machine	Instructions without compatibility with Documentum Process Engine	Instructions with compatibility with Documentum Process Engine
Microsoft Windows	Installing D2 without Documentum Process Engine Compatibility on Microsoft Windows, page 15	Installing D2 with Documentum Process Engine Compatibility on Microsoft Windows, page 19
A Linux environment	Installing D2 without Documentum Process Engine Compatibility on a Linux Environment, page 17	Installing D2 with Documentum Process Engine Compatibility on a Linux Environment, page 20

   d. If you are installing D2 on a Content Server cluster, install the D2 API libraries on each instance of Content Server.
   e. Configure lockbox for each instance of Content Server and create the `D2Method.passphrase`. D2 uses lockbox to manage login tickets. Configuring Lockbox on the Documentum Content Server, page 23 contains further instructions.
   f. Configure the Content Server as described in the following table:

Configuration	Instructions
Configuring `logback.xml`	Configuring logback.xml for the Content Server, page 66
Configuring the display of tables	Configuring Content Server Table Display, page 67
Configuring D2 auditing	Configuring D2 Auditing, page 68

g. Restart Documentum Content Server and JMS.

2. On a system with Documentum Composer, follow the instructions for deploying `D2-DAR.dar`, `D2Widget-DAR.dar` and `Collaboration_Services.dar`. Installing D2 DAR, page 26 contains further instructions.
3. On the web application server:
   a. Stop web application server services.
   b. Follow the instructions for creating the D2 WAR files for deployment as described in the following table:

Operating system of your web application server machine	Instructions
Microsoft Windows	Creating D2 WAR Files on a Web Application Server Using Microsoft Windows, page 29
A Linux environment	Creating D2 WAR Files on a Web Application Server Using a Linux environment, page 31

   c. 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, page 37
Apache Tomcat for a Linux environment	Installing D2 on Apache Tomcat for a Linux Environment, page 38
IBM WebSphere	Installing D2 on IBM WebSphere, page 39
BEA WebLogic	Installing D2 on Oracle WebLogic, page 41
Redhat JBOSS	Installing D2 on Redhat JBOSS, page 42
VMware vFabric tcServer for Microsoft Windows	Installing D2 on VMware vFabric tcServer for Microsoft Windows, page 43

   d. Start application server services.
   e. Configure lockbox for each web application server and create the `D2Method.passphrase`. D2 uses lockbox to manage login tickets. Configuring Lockbox on the Web Application Server, page 44 contains further instructions.
4. Configure D2:
   a. Configure the applications as described in the following table:

Application	Instructions
D2 Config	Configuring D2 Config, page 55
D2 Client	Configuring D2 Client, page 58
D2 Java Method Server	Configuring D2 Java Method Server, page 61

b. Store passwords in the lockbox file. Storing Passwords in the D2 Lockbox, page 61 contains further information and instructions.

5. 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), page 75
Kerberos	Configuring Kerberos, page 76

6. You can configure the file transfer mode. File Transfer Modes, page 64 contains more information about the Java and thin client file transfer modes and Configuring File Transfer Modes, page 64 contains configuration instructions.
7. Install the language packs. Installing D2 Config Language Packs (French-only), page 64 and Installing D2 Client Language Packs, page 65 contain further instructions.
8. Run the applications.
9. Log in to D2 Config and import configurations. The *EMC 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.

Chapter 2

Installing D2 on the Content Server

This chapter contains the following topics:

Installing D2 without Documentum Process Engine Compatibility on Microsoft Windows

1. 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\<user>\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.
2. On the **Select installation packages** page, select **D2-API for Content Server\JMS** and **DAR**.
3. 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.

   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy 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 must 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.

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

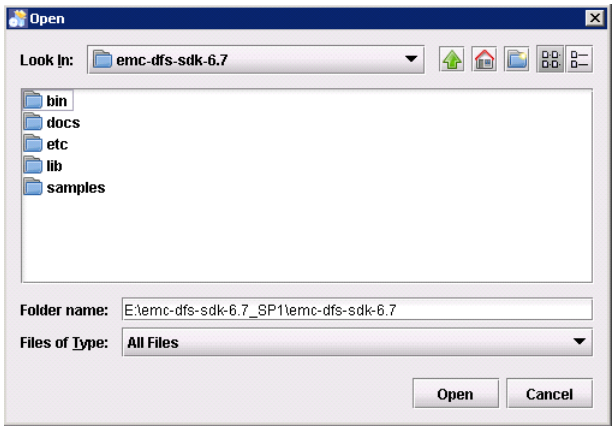
Field	Path
For Content Server	*<install path of D2>*
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>**<JBOSS version>*\DctmServer_Method-Server\deployments\ServerApps.ear For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>**<JBOSS version>*\DctmServer_Method-Server\deploy\ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>**<JBOSS version>*\DctmServer_Method-Server\deploy\ServerApps.ear\APP-INF

5. On the **DAR extraction folder** page, select the path to extract the DAR files.

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

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

7. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

   D2 uses Lockbox to manage login tickets.

8. Read the extraction summary and click **Next**, then click **Done**.
9. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 without Documentum Process Engine Compatibility on a Linux Environment

1. Launch D2 installer from XWindows interface using the owner account for the Documentum installation.
2. 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.
3. On the **Select installation packages** page, select **D2-API for Content Server/JMS** and **DAR**.
4. 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.

   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy 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 must 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.
5. Fill out the **D2-API extraction folders** page.
   a. If your existing Content Server install does not use a `DOCUMENTUM_SHARED` directory, fill out the form as described in the following table:

Field	Path
For Content Server	*<install path of D2>*
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_MethodServer/deployments/ServerApps.ear For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_MethodServer/deploy/ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_MethodServer/deploy/ServerApps.ear/APP-INF

   b. If your existing Content Server install uses a `DOCUMENTUM_SHARED` directory, fill out the form as described in the following table:

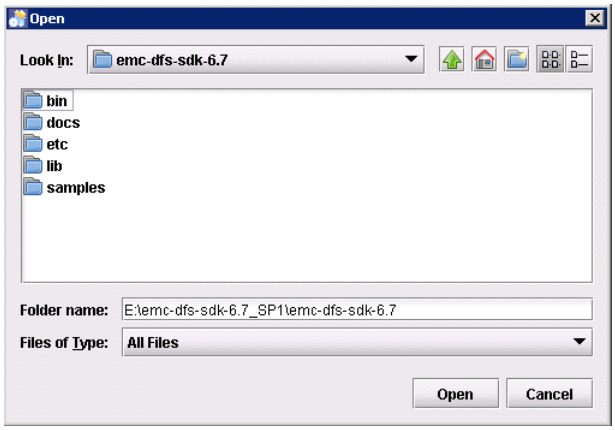
Field	Path
For Content Server	*<install path of D2>*
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_MethodServer/deployments/ServerApps.ear For Content Serversion Version 7.0, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_MethodServer/deploy/ServerApps.ear For Content Server Version 6.7 SP1 and SP2, use *<install path of Documentum>*/shared/*<JBOSS version>*/Dctm-Server_MethodServer/deployment/ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/shared/*<JBOSS version>*/Dctm-Server_MethodServer/deployment/Server-Apps.ear/APP-INF

6. On the **DAR extraction folder** page, select the path to extract the DAR files.

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

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

8. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

   D2 uses Lockbox to manage login tickets.

9. Read the extraction summary and click **Next**, then click **Done**.

10. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 with Documentum Process Engine Compatibility on Microsoft Windows

1. 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\<user>\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.

2. On the **Select installation packages** page, select **D2-API for Content Server\JMS**, **D2–API for BPM**, and **DAR**.
3. 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.

   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy 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 must 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.

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

Field	Path
For Content Server	*<install path of D2>*
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>**<JBOSS version>*\server\DctmServer_MethodServer\deployments\ServerApps.ear For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>**<JBOSS version>*\server\DctmServer_MethodServer\deploy\ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>**<JBOSS version>*\server\DctmServer_MethodServer\deploy\ServerApps.ear\APP-INF
For Business Process Manager	For Content Server Version 7.1 and later, use *<install path of Documentum>**<JBOSS version>*\DctmServer_MethodServer\deployments\bpm.ear

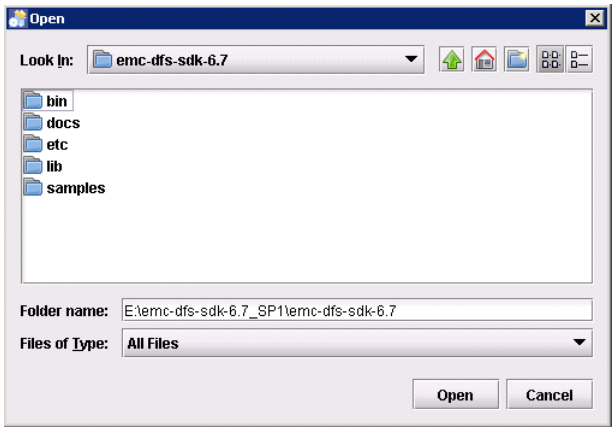
Field	Path
	For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>**<JBOSS version>*\DctmServer_Method-Server\deploy\bpm.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>**<JBOSS version>*\DctmServer_Method-Server\deploy\bpm.ear\APP-INF

5. On the **DAR extraction folder** page, select the path to extract the DAR files.

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

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

7. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

   D2 uses Lockbox to manage login tickets.

8. Read the extraction summary and click **Next**, then click **Done**.
9. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 with Documentum Process Engine Compatibility on a Linux Environment

1. Launch D2 installer from the XWindows interface using the owner account for the Documentum installation.
2. 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.

3. On the **Select installation packages** page, select **D2-API for Content Server/JMS**, **D2–API for BPM**, and **DAR**
4. 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.

   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy 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 must 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.

5. Fill out the **D2-API extraction folders** page.
   a. If your existing Content Server install does not use a `DOCUMENTUM_SHARED` directory, fill out the form as described in the following table:

Field	Path
For Content Server	*<install path of D2>*
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_MethodServer/deployments/ServerApps.ear For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_MethodServer/deploy/ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_MethodServer/deploy/ServerApps.ear/APP-INF
For Business Process Manager	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_Method-Server/deployments/bpm.ear For Content Server Version 6.7 SP1 to 7.0, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_Method-Server/deploy/bpm.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_Method-Server/deploy/bpm.ear/APP-INF

   b. If your existing Content Server install uses a `DOCUMENTUM_SHARED` directory, fill out the form as described in the following table:

Field	Path
For Content Server	*<install path of D2>*

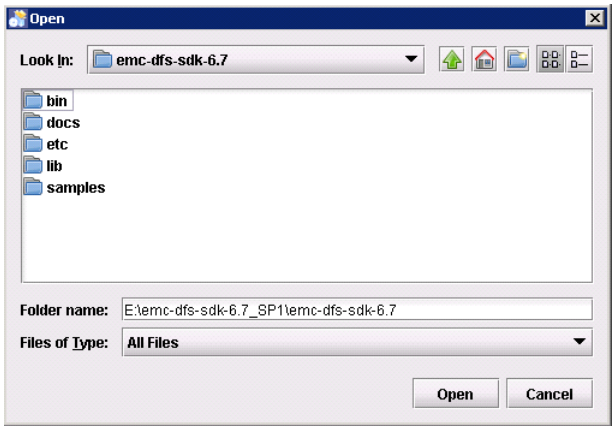
Field	Path
For Java Method Server	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_MethodServer/deployments/ServerApps.ear For Content Serversion Version 7.0, use *<install path of Documentum>*/*<JBOSS version>*/server/Dctm-Server_MethodServer/deploy/ServerApps.ear For Content Server Version 6.7 SP1 and SP2, use *<install path of Documentum>*/shared/*<JBOSS version>*/server/Dctm-Server_MethodServer/deployment/ServerApps.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/shared/*<JBOSS version>*/server/Dctm-Server_MethodServer/deployment/Server-Apps.ear/APP-INF
For Business Process Manager	For Content Server Version 7.1 and later, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_Method-Server/deployments/bpm.ear For Content Serversion Version 7.0, use *<install path of Documentum>*/*<JBOSS version>*/DctmServer_Method-Server/deploy/bpm.ear For Content Server Version 6.7 SP1 and SP2, use *<install path of Documentum>*/shared/*<JBOSS version>*/Dctm-Server_MethodServer/deployment/bpm.ear For Content Server Version 6.7 and earlier, use *<install path of Documentum>*/shared/*<JBOSS version>*/Dctm-Server_MethodServer/deployment/bpm.ear/APP-INF

6. On the **DAR extraction folder** page, select the path to extract the DAR files.

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

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

8. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

   D2 uses Lockbox to manage login tickets.

9. Read the extraction summary and click **Next**, then click **Done**.
10. Return to the instructions: Instructions for Installing D2, page 11.

Configuring Lockbox on the Documentum Content Server

1. Set environment variables for the Lockbox dependency files extracted by the D2 installer.
   a. Add the `D2.jar`, `LB.jar`, and `LBJNI.jar` files extracted by the installer to the `ClassPath` environment variable:
      - *Microsoft Windows*:

        ```
        set CLASSPATH=<D2-API install path>\D2.jar;<D2-API install path>\LB.jar;<D2-API install path>\LBJNI.jar;%ClassPath%
        ```

      - *Linux environments*:`export CLASSPATH <D2-API install path>/D2.jar;<D2-API install path>/LB.jar:/<D2-API install path>/LBJNI.jar:$CLASSPATH`

   b. Add the Lockbox architecture, which is found in the platform-specific folders extracted by the installer, to the environment variables. Refer to the README file found in each platform folder to determine which folder corresponds to your environment. For example, the README file in `win_vc80_x64` lists compatibility with *Microsoft Windows Server 2003 x86_64 (64-bit)*, *Microsoft Windows Server 2008 x86_64 (64-bit)*, and *Microsoft Windows Server 2012 x86_64 (64-bit)*.

      The architecture for the Lockbox file must match the Java Virtual Machine. For example, when you are using *Linux x64* with a 32-bit Java Virtual Machine, you must reference the `linux_ia32` architecture. When you are using an x86 environment (including non-itanium), you must use the ia32 kits.

      - *Microsoft Windows*:`set PATH=<path to the Lockbox files>\<platform folder>;%PATH%`
      - *Linux* environments:

        ```
        export LD_LIBRARY_PATH <path to the Lockbox files>:$LD_LIBRARY_PATH
        ```

        ```
        export PATH <path to the Lockbox files>/<platform folder>:$PATH
        ```

      - *IBM AIX*:

        ```
        export LIBPATH=<path to the Lockbox files>:$LIBPATH
        ```

        ```
        export PATH <path to the Lockbox files>/<platform folder>:$PATH
        ```

   c. Install the Microsoft Visual C++ Redistributable Package MFC Security Update. The platform folder used in step 1b contains the installer file.

d. Copy the *LB.jar* and *LBJNI.jar* files from the Lockbox folder to <install path of Documentum>\<JBoss version>\server\DctmServer_MethodServer\lib folder.

Note: The above step is only needed for Documentum Content Server versions prior to 7.1. If you have Documentum Content Server version 7.1 or later, please proceed to step 5 below.

2. Generate the Lockbox file and set the `D2Method.passphrase` using following command:

```
java com.emc.common.java.crypto.SetLockboxProperty <path to store
D2.Lockbox>D2Method.passphrase<unencrypted password><optional
Lockbox passphrase>
```

Note: The `D2Method.passphrase` must be identical across all physical machines.

When the `D2.Lockbox` file does not exist, when you run the command D2 creates the file for you. You must generate the Lockbox file for each physical machine in your environment, and you must not copy the Lockbox file from one machine to another.

Note: Do not copy the Lockbox file from one machine to another. These Lockbox files must be individually generated for each machine to avoid issues.

When the `D2.Lockbox` file already exists in the *<path to store D2.Lockbox>* folder, the command does not change the passphrase. To change the Lockbox passphrase, you must delete the existing `D2.Lockbox` file.

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

Parameter	Description
<*path to store D2.Lockbox*>	Set the path to store `D2.Lockbox`. The folder must exist before running the command.
<*unencrypted password*>	Set a plaintext password consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter). The D2Method passphrase must be identical across all Document Content Servers, web application servers, and ACS/BOCS servers. **Note**In a Linux environment, the password cannot start with a special character. Rest password rules apply as is.
<*optional Lockbox passphrase*>	You can optionally set a plaintext passphrase consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter) for the Lockbox file. You must use this Lockbox passphrase when the system fails and the Lockbox needs to be recovered from a different server. Optionally, you can create a passphrase using the Java random passphrase generator by using the following command: `Java com.emc.common.java.crypto .PassphraseGenerator <length>` If you do not set a Lockbox passphrase, D2 generates a random passphrase when creating `D2.Lockbox`.

3. Copy the `D2.Lockbox` file generated to the Java Method Server `classes` folder as described in the following table:

Documentum Content Server Version	Path
Documentum Content Server 7.1	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deployments/ServerApps.ear/APP-INF/classes/
Documentum Content Server 6.7 SP1 to 7.0	*<install path of Documentum>*/*<JBoss version>*/DctmServer_MethodServer/deploy/ServerApps.ear/APP-INF/classes/
Documentum Content Server 6.7 and earlier	*<install path of Documentum>*/*<JBoss version>*/DctmServer_MethodServer/deploy/ServerApps.ear/APP-INF/classes/

4. Set up the Lockbox file for Documentum Process Engine:
   a. Copy the `D2.Lockbox` file generated to the Java Method Server `classes` folder as described in the following table:

Documentum Content Server Version	Path
Documentum Content Server 7.1	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deployments/bpm.ear/APP-INF/classes/
Documentum Content Server 6.7 to 7.0	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deploy/bbpm.ear/APP-INF/classes/

   b. Add the `D2.Lockbox` file copied to Documentum Process Engine to the ClassPath environment variable:

      ```
      export CLASSPATH=<path to the Documentum Process EngineD2.Lockbox>/D2.Lockbox:$CLASSPATH
      ```

5. If you are running Documentum Content Server 7.1:
   a. Create the folder *<install path of Documentum>*/*<JBoss version>*/modules/emc/d2/Lockbox/main/
   b. Copy the `LB.jar` and `LBJNI.jar` files extracted by the installer to the `/main/` folder.
   c. Create `module.xml` as described below and save it in the `/main/` folder.

      ```
      <module xmlns="urn:jboss:module:1.1" name="emc.d2.Lockbox">

      <resources>

      <resource-root path="LB.jar" />

      <resource-root path="LBJNI.jar" />

      </resources>

      </module>
      ```

   d. Open *<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deployments/ServerApps.ear/META-INF/jboss-deployment-structure.xml in a text editor.
   e. Add the following lines to the `<deployment>` section:

```
<dependencies>
<module name="emc.d2.Lockbox" />
</dependencies>
```

6. If you are running Documentum Content Server 7.1 and installed compatibility with Documentum Process Engine:
   a. Open the Documentum Process Engine *<install path of Documentum>**<JBoss version>*\server\DctmServer_MethodServer\deployments\bpm.ear\META-INF\jboss-deployment-structure.xml in a text editor.
   b. Add the following lines to the <deployment> section:

```
<dependencies>
<module name="emc.d2.Lockbox" />
</dependencies>
```

7. Return to the instructions: Instructions for Installing D2, page 11.

Deploying DAR Files

You must deploy DAR files during the D2 installation as described in the following table:

DAR	Description
D2-DAR.dar	Deploy to install D2.
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 SP1 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.

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 Content 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.

Field	Description
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Content 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 Content Server installation owner is not named dmadmin, as described in Step 3.

3. If the Content 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, page 11.

Chapter 3

Creating D2 WAR Files

This chapter contains the following topics:

Creating D2 WAR Files on a Web Application Server Using Microsoft Windows

1. Configure Microsoft Internet Explorer settings to avoid interference with D2 as follows:
   - Allow popup windows.
   - Allow windows to resize by script without size or position constraints.
   - Allow the browser to use tabbed browser settings when encountering a popup window.
   - Make D2 URLs a part of the intranet or Trusted Security Zone to allow D2 ActiveX controls and MSXML.
2. 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\<user>\Local Settings\Temp\D2-Installer_x`

   The temporary directory holds the installation logs.
3. On the **Select installation packages** page, select **D2** and **D2-Config**.
4. On the **WebApp extraction folder** page, select a folder to which the installer extracts `D2-Config.war` and `D2.war`. The selected folder must not already contain `D2-Config.war` and `D2.war`.

   If you are using a JBoss 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 web application server, do not select the checkbox. Selecting it causes the the installer removes files required for D2.
5. On the **Configuration file(s) settings** page, configure whether to import configuration files and how you want to deploy the configuration files:
   a. Select the Configuration file(s) source as described in the following table:

Option	Description
Create new Configuration file(s)	Select to create and use a new set of configuration files.
Use existing Configurations file(s) from a specific location	Select to import and use an existing set of configuration files. Click **Browse**, then navigate to and select the set of configuration files.

b. Select the method of deployment as described in the following table:

Option	Description
Include Configuration file(s) into WAR(s)	You can edit the D2 Config configuration files prior to inclusion in the WAR file for deployment. Configuring D2 Config, page 55 contains further information. You can also edit the configuration files after deployment. Select to automatically package configuration files with the WAR deployment.
Extract Configuration file(s) at a specific location	Select to extract configuration files to a specified location. Use this option if you do not want to store configuration files in their default locations, for example to centralize configuration files. Click **Browse**, then navigate to and select the folder you want to use as the configuration file storage location.

6. 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.

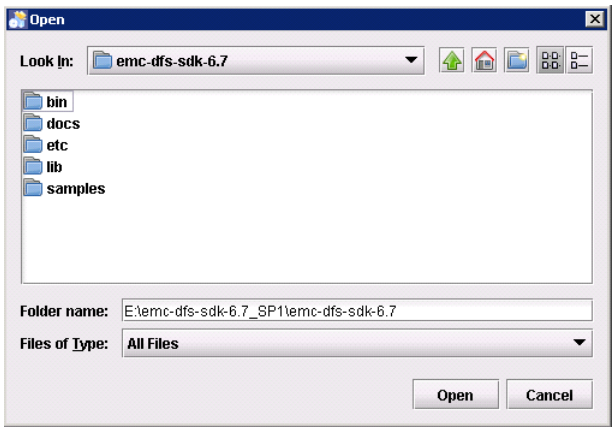
   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy DAR files. For example, if you add `C2-Install.jar`:

   - 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 must 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.

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

8. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

   D2 uses Lockbox to manage login tickets.

9. Read the extraction summary and click **Next**, then click **Done**.
10. Return to the instructions: Instructions for Installing D2, page 11.

Creating D2 WAR Files on a Web Application Server Using a Linux Environment

1. Set the graphical environment, by either:
   - 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 JVM launching the JMS.
2. Launch D2 installer from the XWindows interface using the owner account for the Documentum installation.
3. 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.

4. On the **Select installation packages** page, select **D2** and **D2-Config**.
5. On the **WebApp extraction folder** page, select a folder to which the installer extracts `D2-Config.war` and `D2.war`. The selected folder must not already contain `D2-Config.war` and `D2.war`.

   If you are using a JBoss 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 web application server, do not select the checkbox. Selecting it causes the the installer removes files required for D2.

6. On the **Configuration file(s) settings** page, configure whether to import configuration files and how you want to deploy the configuration files:
   a. Select the Configuration file(s) source as described in the following table:

Option	Description
Create new Configuration file(s)	Select to create and use a new set of configuration files.
Use existing Configurations file(s) from a specific location	Select to import and use an existing set of configuration files. Click **Browse**, then navigate to and select the set of configuration files.

   b. Select the method of deployment as described in the following table:

Option	Description
Include Configuration file(s) into WAR(s)	You can edit the D2 Config configuration files prior to inclusion in the WAR file for deployment. Configuring D2 Config, page 55 contains further information. You can also edit the configuration files after deployment. Select to automatically package configuration files with the WAR deployment.
Extract Configuration file(s) at a specific location	Select to extract configuration files to a specified location. Use this option if you do not want to store configuration files in their default locations, for example to centralize configuration files. Click **Browse**, then navigate to and select the folder you want to use as the configuration file storage location.

7. 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.

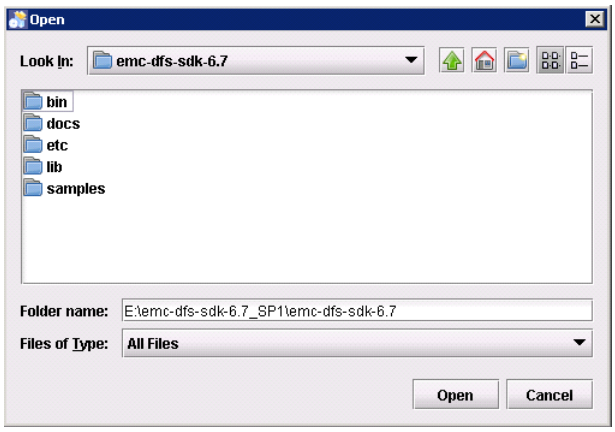
   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy DAR files. For example, if you add `C2-Install.jar`:

   - 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 must 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.

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc—dfs-sdk-6.7` folder shown in the following image:



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

10. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

    D2 uses Lockbox to manage login tickets.

11. Read the extraction summary and click **Next**, then click **Done**.
12. Return to the instructions: Instructions for Installing D2, page 11.

Creating D2 WAR Files on a Web Application Server Using IBM AIX

1. Set the graphical environment, by either:
   - 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 JVM launching the JMS.
2. Launch D2 installer from the XWindows interface using the owner account for the Documentum installation.
3. 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.

4. On the **Select installation packages** page, select **D2** and **D2-Config**.
5. On the **WebApp extraction folder** page, select a folder to which the installer extracts `D2-Config.war` and `D2.war`. The selected folder must not already contain `D2-Config.war` and `D2.war`.

   If you are using a JBoss 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 web application server, do not select the checkbox. Selecting it causes the the installer removes files required for D2.

6. On the **Configuration file(s) settings** page, configure whether to import configuration files and how you want to deploy the configuration files:
   a. Select the Configuration file(s) source as described in the following table:

Option	Description
Create new Configuration file(s)	Select to create and use a new set of configuration files.
Use existing Configurations file(s) from a specific location	Select to import and use an existing set of configuration files. Click **Browse**, then navigate to and select the set of configuration files.

   b. Select the method of deployment as described in the following table:

Option	Description
Include Configuration file(s) into WAR(s)	You can edit the D2 Config configuration files prior to inclusion in the WAR file for deployment. Configuring D2 Config, page 55 contains further information. You can also edit the configuration files after deployment. Select to automatically package configuration files with the WAR deployment.
Extract Configuration file(s) at a specific location	Select to extract configuration files to a specified location. Use this option if you do not want to store configuration files in their default locations, for example to centralize configuration files. Click **Browse**, then navigate to and select the folder you want to use as the configuration file storage location.

7. 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.

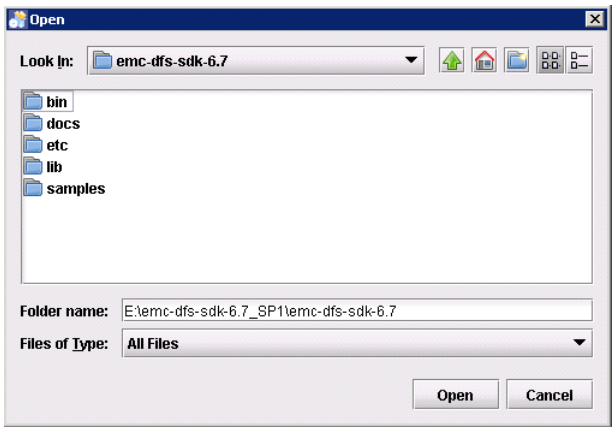
   The D2 Installer automatically runs the added plug-in installer .jar files and activates the plug-in for D2, but you must manually deploy DAR files. For example, if you add `C2-Install.jar`:

   - 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 must 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.

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

   Make sure you select the main DFS SDK folder that contains the `lib` folder, such as the default `emc-dfs-sdk-6.7` folder shown in the following image:



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

10. On the **Lockbox extraction folder** page, select the path to extract the RSA CST Lockbox dependency files.

    D2 uses Lockbox to manage login tickets.

11. Read the extraction summary and click **Next**, then click **Done**.

12. Return to the instructions: Instructions for Installing D2, page 11.

Chapter 4

Installing D2 on the Web Application Server

This chapter contains the following topics:

Installing D2 on Apache Tomcat for Microsoft Windows

1. If you are upgrading D2, do the following. Otherwise, skip to step 2.
   a. Delete the `<install path of Tomcat>\webapps\D2-Config` and `<install path of Tomcat>\webapps\D2` folders.
   b. Clear the Catalina cache in the folder `<install path of Tomcat>\work\Catalina\localhost\`
   c. Clear the Tomcat temporary folder `<install path of Tomcat>\temp\`
2. Copy `D2-Config.war` and `D2.war` to the `<install path of the web application server>\webapps` folder.
3. 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 `<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`

4. Copy `LB.jar` and `LBJNI.jar` to the shared `<install path of Tomcat>\lib` folder.
5. 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.
6. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 on Apache Tomcat for a Linux Environment

1. If you are upgrading D2, do the following. Otherwise, skip to step 2.
   a. Delete the `<install path of Tomcat>/webapps/D2-Config` and `<install path of Tomcat>/webapps/D2` folders.
   b. Clear the Catalina cache in the folder `<install path of Tomcat>/work/Catalina/localhost/`.
   c. Clear the Tomcat temporary folder `<install path of Tomcat>/temp/`
2. Copy `D2-Config.war` and `D2.war` to the `<install path of the web application server>/webapps` folder.
3. 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`

4. Copy `LB.jar` and `LBJNI.jar` to the shared `<install path of Tomcat>/lib/` folder.
5. 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.
6. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 on IBM WebSphere

1. If you are upgrading D2, do the following. Otherwise, skip 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. If you are installing on a cluster:
    1. Click **Next** to show the **Map modules to server** page.
    2. Select the `D2-Config.war` module.
    3. 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.
    4. Click **Apply**, then click **Next** to show the **Map virtual hosts for Web modules** page.
    5. Select the `D2-Config.war` module and click **Next** to show the **Summary** page.
    6. Review the installation summary. You can click **Previous** to modify any selections.
    7. 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 **Application class loader first** mode. The default is the **Parent class loader first** 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 **Application class loader first** mode. The default is the **Parent class loader first** 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. If the configuration files were 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. Add a new shared library for `aspectjrt.jar:`
    a. In the WebSphere Administration console, navigate to **Environment** > **Shared libraries**.
    b. Select a cell for **Scope**.
    c. Click **New** and fill out the form as described in the following table:.

Field	Description
Name	Type `aspectjrt.jar`
Classpath	Type *<install path to WebSphere>*`\AppServer\profiles\AppSrv01\installedApps\`*<cell>*`\D2.ear\D2.war\WEB-INF\lib\aspectjrt.jar`

d. In the **Class loader order** section, click **Use an isolated class loader for this shared library**.

e. Click **OK** and **Save**.

8. Copy all files with the `.xsd` extension from *<install path to WebSphere>*`\AppServer\profiles\AppSrv`*<version>*`\installedApps\`*<Cell>*`\D2.ear\D2.war\WEB-INF\schemas` to *<install path to WebSphere>*`\AppServer\profiles\AppSrv`*<version>*.
9. Verify that `wstx-lgpl-3.2.9.jar` is in *<install path to WebSphere>*`\AppServer\profiles\AppSrv`*<version>*`\installedApps\`*<Cell>*`\D2.ear\D2.war\WEB-INF\lib`
10. If 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.
11. If you are installing D2 on a Linux environment:
    a. Navigate to the `${jboss.home.dir}/common/lib` folder.
    b. Copy `log4j.jar` to the following folders:
       - *<install path to WebSphere>*`/D2-Config.war/WEB-INF/lib`
       - *<install path to WebSphere>*`/D2.war/WEB-INF/lib`
    c. Navigate to the *<install path to WebSphere>*`/D2.war/WEB-INF/lib` folder.
    d. Delete or remove `jsr173_api.jar`.
12. Set up lockbox:
    a. Copy `LB.jar` and `LBJNI.jar` to the *<install path to Websphere>*`\AppServer\lib\` folder.
    b. Add a new shared library for LB.jar and LBJNI.jar, as described in Step 7.
13. Return to the instructions: Instructions for Installing D2, page 11.

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. If the Oracle WebLogic application server is running on Red Hat Enterprise Linux, extract `D2-Config.war` to a temporary folder.

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. Click **Install**, click **Browse**, then select `D2.war`, and click **Active Change**.
   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/plug-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. If you are installing D2 on a Linux environment:
    a. Navigate to the `${jboss.home.dir}/common/lib` folder.
    b. Copy `log4j.jar` to the following folders:
       - *<install path to WebLogic>*`/D2-Config.war/WEB-INF/lib`
       - *<install path to WebLogic>*`/D2.war/WEB-INF/lib`
    c. Navigate to the *<install path to WebLogic>*`/D2.war/WEB-INF/lib` folder.
    d. Delete or remove `jsr173_api.jar`.
11. Copy `LB.jar` and `LBJNI.jar` to the `Oracle/Middlware/user_projects/domains/base_domain/lib` folder.
12. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 on Redhat JBOSS

1. If you are upgrading D2, do the following. Otherwise, skip to step 2.
   a. Delete the *`${jboss.home.dir}`*`/webapps/D2-Config`, *`${jboss.home.dir}`*`/webapps/D2`, and *`${jboss.home.dir}`*`/webapps/D2` folders.
   b. Clear the work folder *`${jboss.home.dir}`*`/server/<jboss instance>/work`

c. Clear the temp folder `${jboss.home.dir}/server/<jboss instance>/tmp`

2. Remove `jboss.vfs.forceVfsJar` or set to `false`.
3. Stop the JBOSS service.
4. Copy and extract `D2-Config.war` to the *<install path of the web application server>*`/webapps/D2-Config.war` folder.
5. Copy and extract `D2.war` to the *<install path of the web application server>*`/webapps/D2.war` folder.
6. If the configuration files were kept out of the WAR files during extraction, copy the configuration files to the `/WEB-INF/classes` folder or configure the references using the classpath definition.
7. Copy `log4j.jar` from the `${jboss.home.dir}/common/lib` folder to the *<install path of the web application server>*`/webapps/D2-Config.war/WEB-INF/lib` folder.
8. Copy `LB.jar` and `LBJNI.jar` to the `/root/jboss-`*<version>*`/jboss-as/server/<jboss instance>/lib/` folder.
9. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 on VMware vFabric tcServer for Microsoft Windows

1. If you are upgrading D2, do the following. Otherwise, skip to step 2.
   a. Delete the *<install path to Tc-server>*`\webapps\D2-Config` and *<install path to Tc-server>*`\webapps\D2` folders.
   b. Clear the cache folder *<install path to Tc-server>*`\logs`
   c. Clear the temp folder *<install path to Tc-server>*`\temp`
   d. Clear the work folder *<install path to Tc-server>*`\work`
2. Copy `D2-Config.war` and `D2.war` to the *<install path of the web application server>*`\webapps` folder.
3. 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.
4. 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"
```

5. Set up lockbox:
   a. Copy `LB.jar` and `LBJNI.jar` to the `\vfabric-tc-server-standard-`*<version>*`\vfabric-tc-server-standard-`*<version>*`.RELEASE\myserver\lib` folder.
   b. Open `wrapper.conf` in a text editor and add the following line to add the lockbox CST path to `java.library.path`:

```
wrapper.java.library.path.<count>=<path to
D2-API>\Lockbox\<platform folder>
```

c. Open `catalina.properties` in a text editor and add the lockbox CST path to `common.loader`.

6. Return to the instructions: Instructions for Installing D2, page 11.

Configuring Lockbox on the Web Application Server

1. Set environment variables for the Lockbox dependency files extracted by the D2 installer.
   a. Add the `C6-Common-<version>.jar`, `LB.jar`, and `LBJNI.jar` files extracted by the installer to the ClassPath environment variable:
      - For Microsoft Windows:

        ```
        set CLASSPATH=<install path of web application
        server>\webapps\D2-Config\WEB-INF\lib\C6-Common-<version>.jar;<path
        to the Lockbox files>\LB.jar;<path to the Lockbox
        files>\LBJNI.jar;%ClassPath%
        ```

      - For Linux environments:

        ```
        export CLASSPATH <install path of web application
        server>/webapps/D2-Config/WEB-INF/lib/C6-Common-<version>.jar:/<path
        to the Lockbox files>/LB.jar:/<path to the Lockbox
        files>/LBJNI.jar:$CLASSPATH
        ```

   b. Add the lockbox architecture found in the platform-specific folders extracted by the installer to the environment variables. Refer to the README found in each platform folder to determine which folder corresponds to your environment. For example, the README in `win_vc80_x64` lists compatibility with Microsoft Windows Server 2003 x86_64 (64-bit), Microsoft Windows Server 2008 x86_64 (64-bit), and Microsoft Windows Server 2012 x86_64 (64-bit).

      The architecture for the lockbox file must match the Java Virtual Machine. For example, if you are using Linux x64 with a 32-bit Java Virual Machine, you must reference the `linux_ia32` architecture. If you are using an x86 environment (including non-itanium), use the ia32 kits.

      - For Microsoft Windows:

        ```
        set PATH=<path to the Lockbox files>\<platform folder>;%PATH%
        ```

      - For Linux environments:

        ```
        export LD_LIBRARY_PATH <path to the Lockbox
        files>:$LD_LIBRARY_PATH
        ```

        ```
        export PATH <path to the Lockbox files>/<platform
        folder>:$PATH
        ```

      - For IBM AIX:

        ```
        export LIBPATH=<path to the Lockbox files>:$LIBPATH
        ```

        ```
        export PATH <path to the Lockbox files>/<platform
        folder>:$PATH
        ```

c. If your web application server uses Microsoft Windows, install the Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package MFC Security Update. The platform folder used in step 1b contains the installer file.

2. Generate the lockbox file and set the `D2Method.passphrase` using following command:

   `java com.emc.common.java.crypto.SetLockboxProperty <path to store D2.lockbox> D2Method.passphrase <unencrypted password> <optional lockbox passphrase>`

   The `D2Method.passphrase` must be identical across all physical machines.

   If `D2.lockbox` does not exist when you run the command, D2 creates the file. You must generate the lockbox file for each physical machine in your environment, and you must not copy the lockbox file from one machine to another.

   If `D2.lockbox` already exists in the *<path to store D2.lockbox>* folder, the command does not change the passphrase. To change the lockbox passphrase, delete the existing `D2.lockbox` file.

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

Parameter	Description
<path to store D2.lockbox>	Set the path to store `D2.lockbox`. The folder must exist before running the command.
<unencrypted password>	Set a plaintext password consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter). The D2Method passphrase must be identical across all Document Content Servers, web application servers, and ACS/BOCS servers. **Note**In a Linux environment, the password cannot start with a special character. Rest password rules apply as is.
<optional lockbox passphrase>	You can optionally set a plaintext passphrase consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter) for the lockbox file. You will use this lockbox passphrase if the system fails and the lockbox needs to be recovered from a different server. Optionally, you can create a passphrase using the Java random passphrase generator by using the following command: `Java com.emc.common.java.crypto.PassphraseGenerator <length>` If you do not set a lockbox passphrase, D2 generates a random passphrase when creating `D2.lockbox`.

3. If you are running Oracle WebLogic, remove `C6-Common-<version>.jar` from the ClassPath environment variable. D2 requires the ClassPath reference to create the lockbox file, but if you do not remove the reference after creating the lockbox file, D2 shows an error on login.
4. Make sure to configure the `lockboxPath` in the following configuration files:

- `D2FS.properties`: Configuring D2 Client, page 58
- `D2-Config.properties`: Configuring D2 Config, page 55

5. Return to the instructions: Instructions for Installing D2, page 11.

Chapter 5

Configuring D2

This chapter contains the following topics:

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.

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

Verify that the host running Documentum Content 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-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
- 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
- LB.jar
- LBJNI.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

Make sure the `D2.jar` file located in this folder is added to the Documentum Content Server CLASSPATH environment variable.

Make sure the .dar files in the `dar` folder are deployed using the Documentum Composer dar deployer.

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

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

- activation.jar
- avalon-framework-<*version*>.jar
- batik-all-<*version*>.jar
- bcmail-jdk<*version*>.jar
- bcprov-jdk<*version*>.jar
- C6-Common.jar
- commons-collections-<*version*>.jar
- commons-io-<*version*>.jar
- commons-lang-<*version*>.jar
- D2-API.jar
- D2-Constants.jar
- D2-Widget-API.jar
- D2FS-Generated.jar
- D2FS4DCTM-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
- LB.jar
- LBJNI.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 Content Server Host

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

- activation.jar
- avalon-framework-<*version*>,jar
- batik-all-<*version*>.jar
- bcmail-jdk<*version*>.jar
- bcprov-jdk<*version*>.jar

- C6-Common.jar
- commons-collections-<*version*>.jar
- commons-io-<*version*>.jar
- commons-lang-<*version*>.jar
- D2-API.jar
- D2FSDCTM-API.jar
- D2FS-Generated.jar
- diff-<*version*>.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
- LB.jar
- LBJNI.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
- 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 `<installation path of web application server>/webapps/` and verify that it contains `D2.war` and `D2-Config.war`.

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

If you are not running a JBoss web application server, verify that `<installation path of web application server>/webapps/D2/WEB-INF/lib/` and `<installation path of web application server>/webapps/D2-Config/WEB-INF/lib/` contain the `log4j.jar` file. If they do not exist and you are not using a JBoss web application server, D2 Config and D2 Client fail to connect to the Docbroker repository.

Verify that the host running the web application 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` and that it contains the `webapps` folder.

D2 Lockbox

After initializing D2.lockbox, log in to D2 Config and D2 Client and navigate to **Help** > **About**. If the **About** dialog box does not show the **D2-API server** version information, you must fix the D2 lockbox installation.

- If D2 throws the `java.lang.IllegalArgumentException: —<error>`, Java Method Server is not able to decrypt the lockbox passphrase. Make sure the `D2Method.passphrase` is identical across all Document Content Servers, web application servers, and BOCS/ACS servers.
- If D2 shows the `NoClassDefFoundError` or `Could not initialize class` errors, make sure you set the CLASSPATH, PATH, or LD_LIBRARY_PATH to the correct CST file for your operating system.

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 Content Server installation uses non-anonymous certificates, copy `dfc.keystore` from the `$dm_home\dba\secure` folder on the Content Server machine. The *EMC Documentum Content Server Installation Guide* contains more information on the Content 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.
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.
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: • 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, page 66 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 Content Server.

Parameter	Description
proxyClientIpHeader	Set to `true` to put the client IP address in the header instead of the proxy IP. Use this setting when you have a proxy in your architecture, as by default the proxy replaces the client IP address with the proxy IP. For example, if disabled, you may not be able to select the correct instance of BOCS.
lockboxPath	Set the parameter to the path of the folder containing the `D2.lockbox` file. For example, `lockboxPath=C:/lockbox_key` Use `/` instead of `\` for the folder path to avoid a cryptography error.

4. Configure `LoadOnStartup` information by setting the following two parameters for each docbase repository in `D2-Config.properties`:

   `LoadOnStartup.<docbase>.username=<installation owner>`

   `LoadOnStartup.<docbase>.domain=<docbase domain>`

   Set the password for the `LoadOnStartup` configuration in lockbox. Storing LoadOnStartup Passwords in the D2 Lockbox, page 61 contains further information on using lockbox.

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, page 11.

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 Content 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, refreshes, or navigates away from D2 in a browser tab or window.
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.

Parameter	Description
connection.remote.url	Uncomment and type the address of the proxy server to enable content transfer in a reverse proxy setup. `http<s>://<proxy or server address>:<port>/D2`
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.

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`

The *D2 4.x Performance Best Practices and Guidelines* white paper contains further information on enabling compression.

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`

Parameter	Description
hideDomain	Set to `true` to hide the domain for login. You can also specify the repository by using the parameter `hideDomain.<repository name>`.
maxResultSetSize	Type a value to limit the result set of all queries used in populating the Users and Groups widgets as well as user and group selection lists in property dialog boxes. Use this parameter to avoid performance problems associated with large result sets. The default value is `1000`.
contentTransferUrlTicketTimeout	Type a value to set the time in minutes that a D2 download URL remains valid.
pluginsOrder	Type a list of plug-ins by name to force the order in which they are loaded. For example, if you have plugin1 that computes data during a property save and plugin2 that verifies data during a property save, you want verification to occur after computation. In this example, set the line as `pluginsOrder=plugin1,plugin2`
lockboxPath	Set the parameter to the path of the folder containing the `D2.lockbox` file. For example, `lockboxPath=C:/lockbox_key` Use / instead of \ for the folder path to avoid a cryptography error.
processXploreResultSet	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`.

6. Configure `LoadOnStartup` information by setting the following two parameters for each docbase repository in `D2FS.properties`:

   `LoadOnStartup.<docbase>.username=<installation owner>`

   `LoadOnStartup.<docbase>.domain=<docbase domain>`

   Set the password for the `LoadOnStartup` configuration in lockbox. Storing LoadOnStartup Passwords in the D2 Lockbox, page 61 contains further information on using lockbox.

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, page 11.

Configuring D2 Java Method Server

1. Stop the JMS.
2. Navigate to the `APP-INF/classes` folder of the Content 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 Security rules to content before applying Autolink, and `false` forces D2 to apply Autolink rules to content before applying Security.
5. Restart the JMS.
6. Return to the instructions: Instructions for Installing D2, page 11.

Storing Passwords in the D2 Lockbox

Store passwords for the D2 web applications in the D2 lockbox file.

D2 Config, D2 Client, and BOCS Servers

On the D2 Config, D2 Client, or BOCS/ACS server, you can set load on startup passwords by using the following command:

```
java com.emc.common.java.crypto.SetLockboxProperty <path to store
D2.lockbox> LoadOnStartup.<docbase>.password <unencrypted password>
```

For example:

```
java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key
LoadOnStartup.Rep1.password ThisIsMyPassword
```

Set the *<path to the D2.lockbox>* and *<unencrypted password>* as described in the following table:

Parameter	Description
<*path to the D2.lockbox*>	Set the path to the `D2.lockbox`. The folder must exist before running the command.
<*unencrypted password*>	Set the plaintext password for the installation owner account consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter).

Single Sign On (SSO)

Set the default password for SSO authentication, either using Microsoft Windows NT Unified Logon or Kerberos, by using the following command:

```
java com.emc.common.java.crypto.SetLockboxProperty <path to store
D2.lockbox> D2FS-trust.<docbase>.password <unencrypted password>
```

You must capitalize `D2FS`. For example:

```
java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key
D2FS-trust.Rep1.password ThisIsMyPassword
```

Set the <*path to the D2.lockbox*> and <*unencrypted password*> as described in the following table:

Parameter	Description
<*path to the D2.lockbox*>	Set the path to the `D2.lockbox`. The folder must exist before running the command.
<*unencrypted password*>	Set the plaintext password for the installation owner account consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter).

Return to Configuring Kerberos section: Configuring Kerberos, page 76.

D2-RPS Connector Plug-in

If you are using the D2-RPS Connector plug-in, store the Java Method Server password on the Documentum Content Server lockbox using the following commands:

```
java com.emc.common.java.crypto.SetLockboxProperty <path to store
D2.lockbox> D2-JMS.<docbase>.password <unencrypted password>
```

```
java com.emc.common.java.crypto.SetLockboxProperty <path to store
D2.lockbox> D2-JMS.<docbase>.<docbase>.password <unencrypted
password>
```

You must run both commands because some applications refer to the repository as <*docbase*>.<*docbase*>. For example, you must run both of the following commands:

- `java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key D2-JMS.Rep1.password ThisIsMyPassword`
- `java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key D2-JMS.Rep1.Rep1.password ThisIsMyPassword`

Set the *<path to the D2.lockbox>* and *<unencrypted password>* as described in the following table:

Parameter	Description
<path to the D2.lockbox>	Set the path to the `D2.lockbox`. The folder must exist before running the command.
<unencrypted password>	Set the plaintext password for the installation owner account consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter).

Upgrading D2 from a Pre-Lockbox Encryption Version

If you are upgrading D2 and stored passwords in the .properties files, you must change the structure of the `LoadOnStartup` parameters in the .properties files and use the `SetLockboxProperty` command using the plaintext version of the password.

For example, if you stored the following parameters in `D2-Config.properties`:

```
LoadOnStartup.docbase.1 = Rep1
LoadOnStartup.username.1 = Admin
LoadOnStartup.domain.1 = MyDomain
LoadOnStartup.password.1 = G8AX4FKASH4L (an encrypted password)
LoadOnStartup.docbase.2 = Rep2
LoadOnStartup.username.2 = dmadmin
LoadOnStartup.domain.2 = Domain2
LoadOnStartup.password = MASD8SG3LKL (an encrypted password)
```

You must use the plaintext version of the password and run the following command:

```
java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key
LoadOnStartup.Rep1.password happy
java com.emc.common.java.crypto.SetLockboxProperty C:/lockbox_key
LoadOnStartup.Rep2.password good
```

You must then replace the `LoadOnStartup` parameters in `D2-Config.properties` with the following:

```
LoadOnStartup.Rep1.username = Admin
LoadOnStartup.Rep1.domain = MyDomain
LoadOnStartup.Rep2.username = dmadmin
LoadOnStartup.Rep2.domain = Domain2
```

If you configured SSO using Microsoft Windows NT Unified Logon or Kerberos, you must remove the *<repository>.<password>* line from `D2FS-trust.properties` after running the `SetLockboxProperty` command.

File Transfer Modes

You can configure how Microsoft Internet Explorer 10 and non-Microsoft Internet Explorer browsers handle file upload and download. D2 offers two options:

- Java applet mode: The default D2 file transfer mode uses the D2 Java applet (X3 Applet) to transfer files. D2 defaults to this configuration if end users are using Microsoft Internet Explorer versions 9 or older to access D2 Client because older versions of Microsoft Internet Explorer do not support HTML5 file upload.
- Thin client mode: Thin client mode uses HTML5 file transfer to download files to a location specified by browser settings. You cannot override browser settings using D2. D2 defaults to this configuration if end users are using the Safari web browser on a Mac iOS to access D2 Client because the Mac iOS does not support Java with Safari.

  Thin client mode hides and disables the native annotation, folder import, and folder export functionality because these features require the Java applet.

Return to the instructions: Instructions for Installing D2, page 11.

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=<mode>`

   Where you can set <*mode*> as:

   - `java`: Java applet mode.
   - `thin`: Thin client mode.
3. If you upgraded the ACS or BOCS server to a version that supports content transfer in thin client mode, 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, page 11.

Installing D2 Config Language Packs (French-only)

1. Install the locale to the repository. The *EMC Documentum Content Server Administration and Configuration Guide* contains future instructions on populating and publishing localized Data Dictionaries into the repository.
2. Download and extract `D2_Config_Language_Packs_<language>_<version>.zip` to a temporary location. This outputs the following files:

- D2-Config-LanguagePack_<*language*>.zip
- D2-RPS-LanguagePack_<*language*>.jar
- D2-Specifications-LanguagePack_<*language*>.jar
- O2-LanguagePack_<*language*>.jar
- D2-Bin-LanguagePack_<*language*>.zip

3. Extract the content of D2-Config-LanguagePack_<*language*>.zip and D2-Bin-LanguagePack_<*language*>.zip to the <*install path to the web application server*>/webapps/D2-Config folder.
4. Copy the language pack for each installed plug-in to the <*install path to the web application server*>/webapps/D2-Config/WEB-INF/lib folder as described in the following table:

Plug-in	File to copy
D2-RPS	D2-RPS-LanguagePack_<*language*>.jar
D2–Specs	D2-Specifications-LanguagePack_<*language*>.jar
O2	O2-LanguagePack_<*language*>.jar

5. Return to the instructions: Instructions for Installing D2, page 11.

Installing D2 Client Language Packs

1. Install the locale to the repository. The *EMC Documentum Content Server Administration and Configuration Guide* contains future instructions on populating and publishing localized Data Dictionaries into the repository.
2. Download and extract D2_Language_Packs_<*language*>_<*version*>.zip to a temporary location. This outputs the following files:
   - D2-LanguagePack_<*language*>.zip
   - C2-LanguagePack_<*language*>.zip
   - D2-Bin-LanguagePack_<*language*>.zip
   - X3Applet-LanguagesPack.jar
3. Extract the content of D2-LanguagePack_<*language*>.zip to the <*install path to the web application server*>/webapps/D2 folder.
4. Navigate to the <*install path to the web application server*>/webapps/D2/applet folder:
   a. Rename or backup the existing X3Applet-LanguagesPack.jar file.
   b. Copy X3Applet-LanguagesPack.jar to the <*install path to the web application server*>/webapps/D2/applet folder.

      If you are installing multiple language packs, you do not need to repeat copying X3Applet-LanguagesPack.jar.
5. If the C2 plug-in is installed, extract the content of C2-LanguagePack_<*language*>.zip to the <*install path to the web application server*>/webapps/D2 folder.

6. If the D2-Bin plug-in is installed, extract the content of `D2-Bin-LanguagePack_<language>.zip` to the `<install path to the web application server>/webapps/D2` folder.
7. Return to the instructions: Instructions for Installing D2, page 11.

Modifying Language Packs

If you do not perform the following steps when modifying the language packs .jar file, D2 shows a Java security popup when an end user logs in to D2 Client.

1. Extract the `X3Applet-LangugesPack.jar`. For example, you can use the following command:

   ```
   jar xf X3Applet-LanguagesPack.jar
   ```

2. Add or modify property resource files in the `resources/i18n` folder.
3. Navigate to and open `META-INF/MANIFEST.MF` and add or set the parameter:

   ```
   Trusted-Only:  true
   ```

4. Rebuild `X3Applet-LanguagesPack.jar`.
5. Sign `X3Applet-LanguagesPack.jar` using a certificate that is trusted by Java Runtime Environment.

   The Oracle Java Tech Notes (http://docs.oracle.com) contain more information about signing .jar files.

6. Deploy the modified and signed `X3Applet-LanguagesPack.jar` to the D2 web application servers.

Configuring logback.xml for the Content Server

Configure logging for:

- The Content 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 Content 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.

3. Return to the instructions: Instructions for Installing D2, page 11.

Removing Debug Logs from the JBoss 7.1.1 JMS Log

1. Navigate to `<install path to Documentum Content Server>/<JBoss version>/server/DctmServer_MethodServer/configuration/` and open `standalone.xml` in a text editor.
2. 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 Content Server server.ini

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

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

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

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

3. 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.

4. Restart the Documentum Content Server.

5. Return to the instructions: Instructions for Installing D2, page 11.

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 Content 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, page 11.

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, page 11.

Disabling xQuery when Using the xPlore Search Engine

By default, the Documentum Foundation Classes search service generates results in xQuery format. This can cause D2 to show results that do not match repository data.

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`
3. Return to the instructions: Instructions for Installing D2, page 11.

Configuring D2 for Documentum Connector for Syncplicity

1. Configure `D2FS.properties` to connect to Documentum Connector for Syncplicity:
   a. Navigate to the location of your D2FS configuration files.

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

   b. Add the line `syncplicityConnectorConfig=`*<connector URL> <connector username> <connector encrypted password>*

      The following table describes the variables:

Variable	Description
connector URL	The URL of the Documentum Connector for Syncplicity.
connector username	The username of the administrator account for Documentum Connector for Syncplicity. Set the username to `CEadmin`.
connector encrypted password	The password of the administration account, as encrypted using the Documentum Connector for Syncplicity encryption tool. The *EMC Documentum Connector for Syncplicity Installation Guide* contains further instructions.

2. Create a profile named `d2-default` in `syncagent.xml`. The *EMC Documentum Connector for Syncplicity Installation Guide* contains instructions for creating a profile.
3. Return to the instructions: Instructions for Installing D2, page 11.

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. If you are using D2-BOCS, install IRM Services for Documentum on each web application server running D2-BOCS.
4. 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.

Chapter 6

Best Practices

This chapter contains the following topics:

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/json"
           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 Content 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.

Chapter 7

Configuring Authentication

This chapter contains the following topics:

Single Sign On (SSO) Authentication for D2

To configure Single Sign On (SSO) authentication for D2, configure the desired form of SSO (NTLM, Kerberos, and TrustedReverseProxy) in the `shiro.ini` file, add the login name of the administrator account for each relevant repository in `d2fs-trust.properties`, add the corresponding administrator passwords to the D2 lockbox. Additionally, configure the `shiro.ini` file 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.

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 the Administrator Login Name to the d2fs-trust.properties file, page 79 the administrator's name to the `d2fs-trust.properties` file.
5. Perform the steps in Adding the Administrator Password to the D2 Lockbox, page 79 to initialize the D2 Lockbox with the administrator password.
6. Perform the steps in Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server, page 79 to configure the `shiro.ini` file.
7. Return to the instructions: Instructions for Installing D2, page 11.

Configuring Kerberos

1. In your Active Directory Server, create a user with the same name as the computer hosting your application server.
   a. Select **Use Kerberos DES encryption types for this account**.
   b. Select **This account supports Kerberos AES 128 bit encryption**.
2. 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. For D2 to work in this scenario, you need to configure an LDAP connection that brings in the users that use both D2 and Webtop. To configure an LDAP connection in DA, set the **user_source** attribute to the empty string in the **Mappings** tab.
3. Perform the steps in Adding the Administrator Login Name to the d2fs-trust.properties, page 79 file to add the administrator's name to the `d2fs-trust.properties` file.
4. Perform the steps in Adding the Administrator Password to the D2 Lockbox, page 79 to initialize the D2 Lockbox with the administrator password.
5. Perform the steps in Configuring the Shiro.ini file for Interoperability with D2-Config and the Documentum Method Server, page 79 to configure the `shiro.ini` file.
6. 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.
7. 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=true

[urls]

##Authentication type

/** = X3-Kerberos
```

8. 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>

   }
   ```

9. 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.
10. 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`

11. 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`.

12. Return to the instructions: Instructions for Installing D2, page 11.

    Refer to Single Sign On (SSO), page 62 section for further information on setting the default password for SSO authentication using Kerberos.

Configuring TrustedReverseProxy

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 `shiro.ini` by adding the following lines:

```
[main]
X3-TrustedReverseProxy =
com.emc.x3.portal.server.filters.authc.X3TrustedReverseProxyHttpAuthenticationFilter
X3-TrustedReverseProxy.userParameterHeaderName=<name of HTTP request
header that holds user login name>

[urls]
##Authentication type
/** = X3-TrustedReverseProxy
```

    For example, if the trusted reverse proxy is CA SiteMinder, then set:
    `X3-TrustedReverseProxy.userParameterHeaderName=SM_USER`

    SSO reverse proxy products such as CA SiteMinder or RSA ClearTrust allow the name of the HTTP request header that will hold the value of the authenticated user's login name to be configured. In the example above for CA SiteMinder, the default name of this header, SM_USER is used.

3. Perform the steps in Adding the Administrator Password to the D2 Lockbox, page 79 to initialize the D2 Lockbox with the administrator password.
4. Perform the steps in Configuring the Shiro.ini file for interoperability with D2-Config and the Documentum Method Server, page 79 to configure the `shiro.ini` file.

Adding the Administrator Login Name to the d2fs-trust.properties file

1. If `d2fs-trust.properties` does not exist, create the file in the webapps `/D2/WEB-INF/classes/` folder using a text editor.
2. Open `d2fs-trust.properties` in the folder `webapps/D2/WEB-INF/classes/` and add the following lines:
   ```
   *.user = <administrator user>
   *.domain = <your domain> [not mandatory]
   #or for each repository
   <repository>.user=<administrator user>
   <repository>.domain=<your domain>
   ```

Adding the Administrator Password to the D2 Lockbox

Adding the administrator password to the D2 Lockbox:

1. Add the administrator's password to the D2 lockbox on the D2 application server using the command-line:
   ```
   java com.emc.common.java.crypto.SetLockboxProperty <path to store
   D2.lockbox> <property name> <unencrypted password> <optional
   lockbox passphrase>
   ```
   The section Configuring Lockbox on the Web Application Server, page 44 provides more details on initializing and updating the lockbox on a D2 application server.
2. If the administrator password is specific to the indicated repository, the property name for this password in the D2 lockbox should be *D2FS-trust.<repository>.password*
3. If the administrator password is the same for all repositories, the property name should be *D2FS-trust.*.password*

   When the D2 application server initializes for SSO, it first looks for the repository specific property name and if it is not found, it looks for the "`D2FS-trust.*.password`" property name in the D2 lockbox.

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 end points 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.

Chapter 8

Installing C2

This chapter contains the following topics:

Installing the C2 Plug-in

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

1. Ensure that the Docbroker and the target repository are running.
2. On the Content Server, extract `C2-Dar-Install.zip` to output the `C2-Dar-Install` folder. This folder contains:

   `C2-DAR.dar` used for installing the C2 DAR.

   `C2-Install.jar` used for installing the C2 plug-in libraries.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `C2-DAR.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Content 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 Content Server installation owner is not named dmadmin, as described in Step 4.

4. If the Content 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 Content Server, JMS services, and web application services.
8. Run C2 Installer on both the Content Server and the application server by following the instructions for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing C2 Libraries on Microsoft Windows, page 82	Installing C2 Libraries on a Linux Environment, page 83

9. Restart the Content 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. Right-click `C2-Install.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\<user>\Local Settings\Temp\C2 x`

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

   The temporary directory holds the installation logs.

2. Click **Browse** to locate and select the folder to which library files are extracted.
3. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `C2-API.jar` and `C2-Plugin.jar`.

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

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deployments\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\APP-INF\lib

You can then delete the extraction folder.

5. 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.
   d. You can then delete the extraction folder.
6. 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`
7. Return to the C2 installation instructions: C2 Overview and Roadmap, page 81.

Installing C2 Libraries on a Linux Environment

1. Launch the C2 installer from the XWindows interface using the owner account for the Documentum installation.
2. Open an xterm and run the installer by typing `java –jar C2-Install.jar`.

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

   `/tmp/C2 `*x*

   Where *x* is the version number.

The temporary directory holds the installation logs.

3. Click **Browse** to locate and select the target installation folder.
4. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `C2-API.jar` and `C2-Plugin.jar`.

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

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deployments/Server-Apps.ear/lib`	`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/lib`	`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/APP-INF/lib`

   You can then delete the extraction folder.

6. 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.
   d. You can then delete the extraction folder.
7. 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`

8. Return to the C2 installation instructions: C2 Overview and Roadmap, page 81.

Chapter 9

Installing O2

This chapter contains the following topics:

Installing the O2 Plug-in

The O2 plug-in adds the ability to manage transferring properties between D2 and Microsoft Office documents. The following table describes file format and software version compatibility:

	Office 97-2003	**Office 2007-2010**	**Open Office 3.2**
.doc	Yes, with macro support	Yes, with macro support	Yes
.docx	No	Yes, with macro support	Yes
.xls	Yes	Yes	No
.xlsx	No	Yes, but cell values must not be empty	No
.ppt	Yes	No	No
.pptx	No	No	No

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

1. Ensure that the Docbroker and the target repository are running.
2. On the Content Server, extract `O2-Dar-Install.zip` to output the `O2-Dar-Install` folder. This folder contains:

   `O2-DAR.dar` used for installing the O2 DAR.

   `O2-Install.jar` used for installing the O2 plug-in libraries.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	**Description**
DAR	Select `O2-DAR.dar`

Field	Description
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Content 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 Content Server installation owner is not named dmadmin, as described in Step 4.

4. If the Content 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 Content Server, JMS services, and web application services.
8. Run O2 Installer on both the Content Server and the application server by following the instructions for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing O2 Libraries on Microsoft Windows, page 86	Installing O2 Libraries on a Linux Environment, page 88

9. Restart the Content 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. Right-click `O2-Install.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\<user>\Local Settings\Temp\O2 x`

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

The temporary directory holds the installation logs.

2. Click **Browse** to locate and select the folder to which library files are extracted.
3. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `O2-API.jar` and `O2-Plugin.jar`.

4. On the Content Server, copy `O2-API.jar` from the extraction folder to the folder depending on your Documentum Content Server version, as described below:

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deployments\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\APP-INF\lib

   You can then delete the extraction folder.

5. 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.
   d. You can then delete the extraction folder.
6. 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`

7. Return to the O2 installation instructions: O2 Overview and Roadmap, page 85.

Installing O2 Libraries on a Linux Environment

1. Launch the O2 installer from the XWindows interface using the owner account for the Documentum installation.
2. Open an xterm and run the installer by typing `java –jar O2-Install.jar`.

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

   `/tmp/O2 x`

   Where *x* is the version number.

   The temporary directory holds the installation logs.
3. Click **Browse** to locate and select the target installation folder.
4. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `O2-API.jar` and `O2-Plugin.jar`.
5. On the Content Server, copy `O2-API.jar` from the extraction folder to the folder depending on your Documentum Content Server version, as described below:

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deployments/Server-Apps.ear/lib`	`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/lib`	`$DOCUMENTUM/<JBOSS version>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/APP-INF/lib`

   You can then delete the extraction folder.
6. 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.
   d. You can then delete the extraction folder.
7. 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`

8. Return to the O2 installation instructions: O2 Overview and Roadmap, page 85.

Chapter 10

Installing D2-Bin

This chapter contains the following topics:

Installing the D2-Bin Plug-in

The D2-Bin plug-in adds recycling bin capabilities to D2 Client. 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 Content Server and application server:

1. Ensure that the Docbroker and the target repository are running.
2. On the Documentum Content Server, extract `D2-Bin-Dar-Install.zip` to output the `D2-Bin-Dar-Install` folder. This folder contains:
   - `D2-Bin-DAR.dar` used for installing the D2-Bin DAR.
   - `D2-Bin-Install.jar` used for installing the D2-Bin plug-in libraries.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `D2-Bin-DAR.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Content 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 Content Server installation owner is not named dmadmin, as described in Step 4.

4. If the Content 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 Content Server, JMS services, and web application services.
8. Run the D2-Bin Installer on both the Content Server and the application server by following the instructions for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing D2-Bin on Microsoft Windows, page 92	Installing D2-Bin on a Linux Environment, page 93

9. Restart the Documentum Content 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. Right-click `D2-Bin-Install.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\<user>\Local Settings\Temp\D2-Bin x`

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

   The temporary directory holds the installation logs.

2. Click **Browse** to locate and select the folder to which library files are extracted.
3. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `D2-Bin-API.jar` and `D2-Bin-Plugin.jar`.

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

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deployments\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\APP-INF\lib

You can then delete the extraction folder.

5. 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.
   d. You can then delete the extraction folder.
6. 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.
7. Return to the D2-Bin installation instructions: D2-Bin Overview and Roadmap, page 91.

Installing D2-Bin on a Linux Environment

1. Launch the D2-Bin installer from the XWindows interface using the owner account for the Documentum installation.
2. Open an xterm and run the installer by typing `java –jar D2-Bin-Install.jar`.

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

   `/tmp/D2-Bin` *x*

   Where *x* is the version number.

   The temporary directory holds the installation logs.
3. Click **Browse** to locate and select the target installation folder.

4. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `D2-Bin-API.jar` and `D2-Bin-Plugin.jar`.

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

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deployments\Server-Apps.ear\lib	$DOCUMENTUM/*<JBOSS version>*/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/lib	$DOCUMENTUM/*<JBOSS version>*/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/APP-INF/lib

   You can then delete the extraction folder.

6. 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.

   d. You can then delete the extraction folder.

7. 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.

8. Return to the D2-Bin installation instructions: D2-Bin Overview and Roadmap, page 91.

Chapter 11

Installing D2 RPS Connector

This chapter contains the following topics:

Installing the D2 RPS Connector Plug-in

The D2 RPS Connector plugin adds a set of PDF control capabilities to D2 Client and a set of configuration modules to D2 Config. The following installation contains the steps for deploying the DAR file and then installing the D2-RPS Connector plug-in library files on both the Documentum Content Server and application server:

1. Ensure that the Docbroker and the target repository are running.
2. On the Content Server, extract `D2-RPS-Dar-Install.zip` to output the `D2-RPS-Dar-Install` folder. This folder contains:

   `D2-RPS.dar` used for installing the D2-RPS Connector DAR.

   `D2-RPS-Install-<version>.jar` used for installing the D2-RPS Connector plug-in libraries.
3. Run the DAR Installer shipped with Documentum Composer, `dardeployer.exe`, and fill out the form as described in the following table:

Field	Description
DAR	Select `D2-RPS.dar`
Docbroker Details	Select the target Docbroker and port. Click **Connect**.
Repository Details	Select the repository with the Content 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 Content Server installation owner is not named dmadmin, as described in Step 4.

4. If the Content 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 Content Server, JMS services, and web application services.
8. Run the RPS Connector installer on both the Content Server and the application server by following the instructions for your operating system as described in the following table:

Microsoft Windows	A Linux environment
Installing D2 RPS Connector Libraries on Microsoft Windows, page 96	Installing D2 RPS Connector Libraries on a Linux Environment, page 98

9. Restart the Content 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. Right-click `D2-RPS-Install-<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\<user>\Local Settings\Temp\D2-RPS x`

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

   The temporary directory holds the installation logs.

2. Click **Browse** to locate and select the folder to which library files are extracted.
3. Follow the instructions until the installation completes, then click **Finish**.

   The installer outputs `D2-RPS-Connector-API.jar` and `D2-RPS-Connector-Plugin.jar`.

4. On the Content Server:

a. Copy the following files to the `lib` folder depending on your Documentum Content Server version, as described below:

- `D2-RPS-Connector-API.jar` from the `D2-RPS-Install-<version>.jar` extraction folder.
- `DmcPolicyEngine.jar` from the EMC Retention Policy Services Administrator (RPSA) webapp.
- `DmcRecords.jar` from the EMC RPSA webapp.
- `DmcRps.jar` from the EMC RPSA webapp.
- `IDmcPolicyEngine.jar` from the EMC RPSA webapp.
- `IDmcRps.jar` from the EMC RPSA webapp.
- `IDmcRpsModules.jar` from the EMC RPSA webapp.
- `emc-policy-services.jar` from the EMC Documentum Retention Policy Services WebServices.
- `emc-retentionmarkup-services.jar` from the EMC Documentum Retention Policy Services WebServices.

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deployments\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\lib	*<install path of Documentum>**<JBOSS version>*\server\Dctm-Server_Method-Server\deploy\Server-Apps.ear\APP-INF\lib

You can then delete the extraction folder.

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 Content Server Version	Path
Documentum Content Server 6.7 SP1 to 7.0	*<install path of Documentum>**<JBoss version>*\server\Dctm-Server_MethodServer\deploy\ServerApps.ear\lib\
Documentum Content Server 6.7 or older	*<install path of Documentum>**<JBoss version>*\server\Dctm-Server_MethodServer\deploy\ServerApps.ear\APP-INF\lib\

c. Store the Java Method Server password in the D2 lockbox. Storing Passwords in the D2 Lockbox, page 61 contains further information.

5. 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.

d. You can then delete the extraction folder.

6. 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`

7. Return to the D2 RPS Connector installation instructions: D2 RPS Connector Overview and Roadmap, page 95.

Installing D2 RPS Connector Libraries on a Linux Environment

1. Launch the D2 RPS Connector installer from the XWindows interface using the owner account for the Documentum installation.
2. Open an xterm and run the installer by typing `java –jar D2-RPS-Install-<version>.jar`.

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

`/tmp/D2-RPS x`

Where *x* is the version number.

The temporary directory holds the installation logs.

3. Click **Browse** to locate and select the target installation folder.
4. Follow the instructions until the installation completes, then click **Finish**.

The installer outputs `D2-RPS-Connector-API.jar` and `D2-RPS-Connector-Plugin.jar`.

5. On the Content Server:

a. Copy the following files to the `lib` folder depending on your Documentum Content Server version, as described below:

- `D2-RPS-Connector-API.jar` from the `D2-RPS-Install-<version>.jar` extraction folder.

- DmcPolicyEngine.jar from the EMC Retention Policy Services Administrator (RPSA) webapp.
- DmcRecords.jar from the EMC RPSA webapp.
- DmcRps.jar from the EMC RPSA webapp.
- IDmcPolicyEngine.jar from the EMC RPSA webapp.
- IDmcRps.jar from the EMC RPSA webapp.
- IDmcRpsModules.jar from the EMC RPSA webapp.
- emc-policy-services.jar from the EMC Documentum Retention Policy Services WebServices.
- emc-retentionmarkup-services.jar from the EMC Documentum Retention Policy Services WebServices.

Documentum Content Server 7.1	Documentum Content Server 6.7 SP1 to 7.0	Documentum Content Server 6.7 and older
<install path of Documentum>/<*JBOSS version*>/server/Dctm-Server_Method-Server/deployments/Server-Apps.ear/lib	<install path of Documentum>/<*JBOSS version*>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/lib	<install path of Documentum>/<*JBOSS version*>/server/Dctm-Server_Method-Server/deploy/Server-Apps.ear/APP-INF/lib

You can then delete the extraction folder.

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 Content Server Version	Path
Documentum Content Server 6.7 SP1 to 7.0	<*install path of Documentum*>/<*JBoss version*>/server/Dctm-Server_MethodServer/deploy/ServerApps.ear/lib/
Documentum Content Server 6.7 or older	<*install path of Documentum*>/<*JBoss version*>/server/Dctm-Server_MethodServer/deploy/ServerApps.ear/APP-INF/lib/

c. Store the Java Method Server password in the D2 lockbox. Storing Passwords in the D2 Lockbox, page 61 contains further information.

6. 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.
   d. You can then delete the extraction folder.
7. 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

8. Return to the D2 RPS Connector installation instructions: D2 RPS Connector Overview and Roadmap, page 95.

Chapter 12

Installing D2-BOCS

This chapter contains the following topics:

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 Content Server. This allows D2 Client to use BOCS for the checking in, importing, and requesting files.

ACS are installed as part of every Content 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).

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 Content 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 Content 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 Content Server and remote end-user locations. The BOCS host machine does not need Content 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 *EMC Documentum Branch Office Caching Services Release Notes* and *EMC Documentum D2-BOCS Installation Guide* contain further instructions.
   b. Ensure Documentum Foundation Services (DFS) is installed on the BOCS server and that the versions match.
   c. To enable asynchronous transfers, ensure Documentum Messaging Services (DMS) is installed on the Content Server.
   d. Ensure Documentum Foundation Classes (DFC) and D2 Client are installed on the application server.
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 Content Server 7.1: Installing D2–BOCS on a BOCS Server for Documentum Content Server 7.1, page 105 For Documentum Content Server 7.0: Installing D2–BOCS on a BOCS Server for Documentum Content Server 7.0, page 104. For Documentum Content Server 6.7 SP2 and older: Installing D2–BOCS on a BOCS Server for Documentum Content Server 6.7 SP2 and Older, page 103.	Installing D2–BOCS on an ACS Server, page 106.

4. Configure D2 Client to enable BOCS. Configuring D2 Client for BOCS, page 107 contains further instructions.
5. Set the network location identifier for the D2-BOCS servers. BOCS and ACS Network Locations, page 118 contains more information and Setting the BOCS and ACS Network Locations, page 119 contains instructions for setting the network location identifier parameter.
6. Configure lockbox for D2-BOCS servers. Configuring Lockbox for BOCS, page 108 contains further instructions.
7. Configure other plug-ins to interact with D2-BOCS as described in the following table:

Plugin	Instructions
O2	Configuring O2 for BOCS, page 112
C2	Configuring C2 for BOCS, page 114

8. You can configure the following optional settings:
   - Enabling BOCS Content Transfer with Non-Anonymous Certificate Based SSL, page 117 contains further instructions on enabling content transfer using BOCS with non-anonymous certificate-based SSL connections.
   - Enabling Compressing for Upload and Download, page 117 contains further information and instructions on enabling file compression for upload and download through D2-BOCS.
   - Enabling Asynchronous BOCS Write, page 118 contains further instructions for enabling asynchronous write.
9. Verify the installation of D2-BOCS and the status of the BOCS or ACS server. Checking D2-BOCS Installation, page 116 contains further instructions.

Installing D2–BOCS on a BOCS Server for Documentum Content Server 6.7 SP2 and Older

1. Ensure Documentum BOCS is installed on a dedicated server. The *EMC Documentum Branch Office Caching Services Release Notes* and *EMC Documentum D2-BOCS Installation Guide* contain further instructions.
2. Ensure the DFS installed on the BOCS server matches versions with BOCS.
3. Download and extract the contents of `D2-BOCS.war` to `<install path of Documentum>/<JBoss version>/server/DctmServer_BOCS/deploy/bocs.ear`

Ensure the extracted folder is named `D2-BOCS.war`, as shown in the following screenshot:

4. Delete the `D2-BOCS.war` file.
5. Navigate to *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/bocs.ear/META-INF/` and open for editing `application.xml`
6. Add the following lines:

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

7. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

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

1. Download and extract the contents of `D2-BOCS.war` to *<install path of Documentum>/<JBoss version>*`/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>/<JBoss version>*`/server/DctmServer_BOCS/deploy/bocs.ear/lib/` to *<install path of Documentum>/<JBoss version>*`/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>*/*<JBoss version>*/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, page 102.

Installing D2–BOCS on a BOCS Server for Documentum Content Server 7.1

1. Download and extract the contents of D2-BOCS.war to *<install path of Documentum>*/*<JBoss version>*/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>*/*<JBoss version>*/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
4. Delete the following files from *<install path of Documentum>*/*<JBoss version>*/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>*/*<JBoss version>*/server/DctmServer_BOCS/deployments/
6. Copy bocs.ear/lib/configs.jar/anonymous-service-handler-chain.xml to D2-BOCS.war/WEB-INF/lib/bocs-ws.jar/com/documentum/acs/ws/ws/

7. Copy `bocs.ear/lib/configs.jar/authorized-service-handler-chain.xml` to `D2-BOCS.war/WEB-INF/lib/D2FS-Generated-4.2.0.jar/com/emc/d2fs/dctm/api/services/ws/`
8. Navigate to *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/`, 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=
      ```

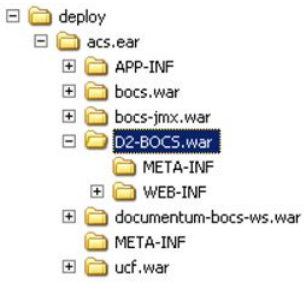
9. Restart the BOCS.
10. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

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>/<JBoss version>*`/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 Content Server version 7.1:
   a. Copy `applicationContext-webservice-client-bocs.xml` from the `configs.jar` folder to the `services/ws/` folder as described in the following table:

Folders	Paths
Original location	*<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/lib/configs.jar/applicationContext-webservice-client-bocs.xml
Destination	*<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/D2FS-Generated-4.2.0.jar/com/emc/d2fs/dctm/api/services/ws/

b. Navigate to *<install path of Documentum>/<JBoss version>*/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 Content Server Version	Path
Documentum Content Server 7.1	*<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/META-INF/
Documentum Content Server 6.7 to 7.0	*<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/META-INF/

   b. Add the following lines:

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

5. Restart the BOCS.
6. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

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 Content 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 communications 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 Content 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, page 102.

Configuring Lockbox for BOCS and ACS Servers

1. On the BOCS server, create a lockbox folder. For example, `C:\Documentum\D2\Lockbox`
2. Copy `C6-Common-<version>.jar`, `LB.jar`, `LBJNI.jar`, and the correct platform folder from the web application server to the lockbox folder you created on the BOCS server.
3. Set environment variables for the Lockbox dependency files extracted by the D2 installer.
   a. Add `C6-Common-<version>.jar`, `LB.jar`, and `LBJNI.jar` to the ClassPath environment variable:
      - For Microsoft Windows:

        ```
        set CLASSPATH=<path to the lockbox folder>\C6-Common-<version>.jar;<path to the lockbox folder>\LB.jar;<path to the lockbox folder>\LBJNI.jar;%ClassPath%
        ```
      - For Linux environments:

```
export CLASSPATH <path to the lockbox
folder>/C6-Common-<version>.jar;<path to the
lockbox folder>/LB.jar:/<path to the lockbox
folder>/LBJNI.jar:$CLASSPATH
```

b. Add the lockbox architecture found in the platform-specific folders extracted by the installer to the environment variables. Refer to the README found in each platform folder to determine which folder corresponds to your environment. For example, the README in `win_vc80_x64` lists compatibility with Microsoft Windows Server 2003 x86_64 (64-bit), Microsoft Windows Server 2008 x86_64 (64-bit), and Microsoft Windows Server 2012 x86_64 (64-bit).

   The architecture for the lockbox file must match the Java Virtual Machine. For example, if you are using Linux x64 with a 32-bit Java Virual Machine, you must reference the `linux_ia32` architecture. If you are using an x86 environment (including non-itanium), use the ia32 kits.

   - For Microsoft Windows:

     ```
     set PATH=<path to the Lockbox files>\<platform folder>;%PATH%
     ```

   - For Linux environments:

     ```
     export LD_LIBRARY_PATH <path to the Lockbox
     files>:$LD_LIBRARY_PATH
     ```

     ```
     export PATH <path to the Lockbox files>/<platform
     folder>:$PATH
     ```

   - For IBM AIX:

     ```
     export LIBPATH=<path to the Lockbox files>:$LIBPATH
     ```

     ```
     export PATH <path to the Lockbox files>/<platform
     folder>:$PATH
     ```

4. If you are using a 64-bit operating system, make sure you are using a 64-bit version of Java Runtime Environment.
5. Generate the lockbox file and set the `D2Method.passphrase` using following command:

   ```
   java com.emc.common.java.crypto.SetLockboxProperty <path to store
   D2.lockbox> D2Method.passphrase <D2Method passphrase> <optional
   lockbox passphrase>
   ```

   The `D2Method.passphrase` must be identical across all physical machines.

   If `D2.lockbox` does not exist when you run the command, D2 creates the file. You must generate the lockbox file for each physical machine in your environment, and you must not copy the lockbox file from one machine to another.

   If `D2.lockbox` already exists in the *<path to store D2.lockbox>* folder, the command does not change the passphrase. To change the lockbox passphrase, delete the existing `D2.lockbox` file.

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

Parameter	Description
<path to store D2.lockbox>	Set the path to store `D2.lockbox`. The folder must exist before running the command.

Parameter	Description
<D2Method passphrase>	Set a plaintext password consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter). The D2Method passphrase must be identical across all Document Content Servers, web application servers, and ACS/BOCS servers. **Note**In a Linux environment, the password cannot start with a special character. Rest password rules apply as is.
<optional lockbox passphrase>	You can optionally set a plaintext passphrase consisting of 8 to 16 characters (at least 1 special character, 1 number, and 1 uppercase letter) for the lockbox file. You will use this lockbox passphrase if the system fails and the lockbox needs to be recovered from a different server. Optionally, you can create a passphrase using the Java random passphrase generator by using the following command: `Java com.emc.com-mon.java.crypto.PassphraseGenera-tor <length>` If you do not set a lockbox passphrase, D2 generates a random passphrase when creating `D2.lockbox`.

6. Set up the lockbox file:
   a. Copy the `D2.lockbox` file generated to the BOCS or ACS `classes` folder as described in the following table:

Server type	Location
BOCS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/classes/` For Content Server 6.7 SP1 to 7.0, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/classes/` For Content Server 6.7, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/bocs.ear/D2-BOCS/WEB-INF/classes/`
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/classes/`

Server type	Location
	For Content Server 6.7 SP1 to 7.0, use *<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/classes/ For Content Server 6.7, use *<install path of Documentum>*/*<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS/WEB-INF/classes/

b. Add the `D2.lockbox` file to the ClassPath environment variable:

- For Microsoft Windows:

  ```
  set CLASSPATH=<path to the BOCS classes folder>\D2.lockbox;%ClassPath%
  ```

- For Linux environments:

  ```
  export CLASSPATH=<path to the BOCS classes folder>/D2.lockbox:$CLASSPATH
  ```

7. On the BOCS server, copy `LB.jar` and `LBJNI.jar` to the `lib` folder of each server as described in the following table:

Documentum Content Server Version	Location
Content Server 7.1	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/
Content Server 6.7 SP1 to 7.0	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/
Content Server 6.7	*<install path of Documentum>*/*<JBoss version>*/server/DctmServer_BOCS/deploy/bocs.ear/D2-BOCS/WEB-INF/lib/

8. If you are using Documentum Content Server 7.1, on ACS servers:

a. Navigate to *<install path of Documentum>**<JBoss version>*\server\DctmServer_MethodServer\deployments\acs.ear\META-INF\ and create or open `jboss-deployment-structure.xml` in a text editor.

b. Create or find the `<sub-deployment name="D2-BOCS.war">` section and add the following lines:

```
<dependencies>

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

</dependencies>
```

9. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

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	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/` For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/Classes/`
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/` For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/`

   Use forward slashes for the file path. For example:

   - To use an absolute path: *<install path of Documentum>/<JBoss version>*`/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	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/

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	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/

4. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

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.

1. Add and set the value of `plugin_x=` to the path for `C2-Plugin.jar` in `D2-BOCS.properties`.

   The location of `D2-BOCS.properties` can differ based on the server type. The following table describes the locations:

Server type	Location
BOCS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/Classes/` For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/Classes/`
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deployments/acs.ear/D2-BOCS.war/WEB-INF/Classes/` For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*`/server/DctmServer_BOCS/deploy/acs.ear/D2-BOCS.war/WEB-INF/Classes/`

   Use forward slashes for the file path. For example:

   - To use an absolute path: *<install path of Documentum>/<JBoss version>*`/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	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/

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	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deployments/D2-BOCS.war/WEB-INF/lib/plugins/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_BOCS/deploy/D2-BOCS.war/WEB-INF/lib/plugins/
ACS	For Content Server 7.1, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deployments/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/ For Content Server 7.0 and older, use *<install path of Documentum>/<JBoss version>*/server/DctmServer_MethodServer/deploy/acs.ear/D2-BOCS.war/WEB-INF/lib/plugins/

4. Return to the BOCS installation instructions: BOCS Pre-requisite and Roadmap, page 102.

Checking D2-BOCS Installation

You can perform the following steps to confirm the state of your BOCS installation.

1. 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.xxxs">
     <plugins />
   </bocs>
   ```

   If you have plug-ins installed, the browser shows the following message:

   ```
   <?xml version="1.0" encoding="utf-8">
   -<bocs version="4.x.x build xxx" server_time="x.xxxs">
     <plugins>C2 v4.x.x build xx</plugins>
   </bocs>
   ```

2. To check for the correct installation of 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.xxxs">
     <plugins />
   </bocs>
   ```

   If you have plug-ins installed, the browser shows the following message:

   ```
   <?xml version="1.0" encoding="utf-8">
   -<bocs version="4.x.x build xxx" server_time="x.xxxs">
     <plugins>C2 v4.x.x build xx</plugins>
   </bocs>
   ```

3. To check the status of your 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, page 102.

Enabling BOCS Content Transfer with Non-Anonymous Certificate Based SSL

1. Enable non-anonymous certificate-based Secure Sockets Layer (SSL) on your Documentum Content Server. The *EMC Documenum Content 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 *EMC Documentum Content Server Distributed Configuration Guide* contains instructions for enabling SSL.
3. Copy `dfc.keystore` from the Content 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, page 102.

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 Content 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, page 102.

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 EMC 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
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. To initialize an expensive precache process of BOCS locations on D2 startup instead of the initial content transfer:
   a. Set the following parameters:

      LoadOnStartup.docbase

      LoadOnStartup.username

      LoadOnStartup.password

      LoadOnStartup.domain

   b. Set the cacheBocsUrl parameter to true in the following two locations:

      - *<install path to web application server>*/webapps/D2/WEB-INF/classes/D2FS.properties
      - *<install path to D2-BOCS>*/WEB-INF/classes/D2-BOCS.properties

Using D2-BOCS for Upload

Do not use D2-BOCS for upload unless you upload large content. The EMC 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.

Chapter 13

Uninstalling D2

This chapter contains the following topics:

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_`
   - All methods added by the D2 DAR, which are prefixed with `D2_`
   - All jobs added by the D2 DAR, which are prefixed with `D2_`
2. On the Content Server:
   a. Remove CLASSPATH references to `D2.jar` and delete the folder containing the D2 files.
   b. 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.

Chapter 14

Troubleshooting the Installation

This chapter contains the following topics:

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 Content Server machine.

Resolution

Use the `SQL ALTER` command to set `READ_COMMITED_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 Content 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 Content 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 Content Server Documentum Foundation Classes.

Unable to Login to D2 After Initializing Lockbox on Oracle Weblogic

Problem

Logging in to D2 shows the error message: `wrong ELF class: ELFCLASS64 (Possible cause: architecture word width mismatch)`

Cause

Mismatched environment with a 64-bit Oracle WebLogic and 32-bit Java Virtual Machine.

Resolution

Use a 64-bit Java Virtual Machine with a 64-bit Oracle WebLogic environment.

Unable to Set Lockbox Property on a 64-bit Operating System

Problem

The `java com.emc.common.java.crypto.SetLockboxProperty <location> D2Method.passphrase <passphrase>` command does not work on a 64-bit operating system.

The command line shows the error:

```
JVM : 1.6.0_27-rev (32bits)
Exception in thread "main" java.lang.NoClassDefFoundError:
Could not initialize class com.emc.clb.clbBridge.clbBridgeJNI
```

Cause

Mismatched environment with a 64-bit operating system and 32-bit Java.

Resolution

Use 64-bit Java with a 64-bit operating system.

Unable to Receive the Task after Using Update Performer Action

Problem

The next user cannot receive the task after using **Update Performer** action

Cause

The way `D2EventSenderMailMethod` works has to adapt to the argument signature of the mailing method in the Content Server layer which is changed due to a security fix.

As part of the security fix, the argument structure of the default mail method of the Content Server (dm_event_sender) was changed wherein, an additional argument web_server_port was added. Before the fix, the web_server_port information was fused with another argument (web_server) using a pipe (|).

Resolution

The dm_event_sender method needs arguments without pipe (|), hence we need to split the web_server argument by removing the pipe and adding an additional argument of web_server_port.

Add an environment variable *addWebServerPort* and set it to `TRUE` in content server environment to force the splitting of the web_server argument.

Note: Above workaround is applicable only for the users who use latest version of Content Sever, if Content Server is not up-to date, then above configuration not required.

Chapter 15

Configuration Files

This chapter contains the following topics:

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.
- d2fs-trust.properties

  If you did not create the file during configuration of authentication, `d2fs-trust.properties` may not exist.

Return to the instructions: Upgrading D2, page 10.