Legal Notice

Copyright © 2000–2017 EMC Corporation. All Rights Reserved.

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** ........................................................................................................................................................................... 9

**Chapter 1**  
Quick Start ............................................................................................................................................................................. 11

**Chapter 2**  
Planning for Deployment .......................................................................................................................... 13

- Typical configuration .............................................................................................................................................. 13
- Preparing the Content Server .......................................................................................................................... 14
- Application server host requirements ............................................................................................................ 15
    - Directory name restriction ......................................................................................................................... 15
    - Content transfer directory permissions ................................................................................................... 15
    - DNS resolution ............................................................................................................................................. 15
- Deploying multiple applications ....................................................................................................................... 15
- Deploying language packs .............................................................................................................................. 16
- Customizing an application ............................................................................................................................ 16

**Chapter 3**  
Preparing the Client Hosts ............................................................................................................................. 17

- Ensuring a certified JVM on browser clients ................................................................................................. 17
- Supporting Outlook mail message .................................................................................................................. 17
    - Using MailApp DAR for email messages ................................................................................................. 18
    - Configuring MailApp.properties .............................................................................................................. 18
    - Using configuration options in app.xml for email import ......................................................................... 19
- Using Citrix Presentation Server Client ........................................................................................................ 19
- Ensuring active scripting is enabled .............................................................................................................. 19
- Using browser extension-based content transfer mechanism ................................................................... 20
- Installing browser extension and native client application ........................................................................ 20
    - Installing in Internet Explorer .................................................................................................................. 20
    - Installing in Firefox ................................................................................................................................... 21
    - Installing in Chrome .................................................................................................................................. 22
    - Locating the installation and log files ........................................................................................................ 22

**Chapter 4**  
Preparing the Application Server Host .............................................................................................................. 25

- Setting the Java memory allocation ............................................................................................................... 25
- Disabling failover ............................................................................................................................................. 25
- Preparing environment variables for non-default DFC locations ................................................................ 26
- Preparing JBoss .................................................................................................................................................. 26
    - Deploying multiple applications on JBoss ................................................................................................. 27
    - Enabling HTTPOnly Cookies Support ....................................................................................................... 29
- Preparing Tomcat ............................................................................................................................................. 29
- Preparing vFabric tc Server ........................................................................................................................... 29
- Preparing WebLogic server ............................................................................................................................. 30
Supporting large content transfer operations in a managed server environment ................................................................. 30
Preparing IBM WebSphere ........................................................................................................................................ 30
Preparing the Application Server for Java 2 security ............................................................................................... 30
Preparing to use an external web server ....................................................................................................................... 31

Chapter 5 Deploying a WDK-based Application ................................................................................................................. 33
Preparing the WAR file for deployment ........................................................................................................................... 33
Enabling DFC connections to repositories ..................................................................................................................... 34
Enabling DFC memory optimization ............................................................................................................................. 35
Configuring UCF ........................................................................................................................................................... 35
Forcing UCF to install a configured JRE ......................................................................................................................... 36
Enabling presets and preferences repositories .............................................................................................................. 36
  Configuring encrypted password for presets and preferences repositories ................................................................. 37
Enabling HttpOnly cookies support for Webtop ............................................................................................................... 38
Configuring to validate application URL ....................................................................................................................... 38
Configuring for improved advanced search .................................................................................................................. 39
Configuring search results ............................................................................................................................................ 39
Configuring to view preferred time zones ...................................................................................................................... 39
Enabling retention of folder structure and objects on export ......................................................................................... 40
Enabling external searches ............................................................................................................................................ 40
  Configuring the connection to the search server ........................................................................................................ 40
  Configuring the connection to the backup search server .......................................................................................... 40
Enabling xPlore search with Webtop emails ................................................................................................................ 41
Deploying multiple applications ..................................................................................................................................... 41
Enabling Documentum Collaborative Services ............................................................................................................ 41
Configuring Client capability roles ..................................................................................................................................... 42
Deploying Documentum Webtop in a cluster for Load Balancing and Failover scenario .................................................. 43

Chapter 6 Completing the Deployment .................................................................................................................................. 45
Configuring IBM WebSphere after deployment .............................................................................................................. 45
  Changing the classloader and compiler settings .......................................................................................................... 45
  Setting custom webcontainer property com.ibm.ws.webcontainer.invokefilterscompatibility to true .................... 46
Deploying default virtual link support ........................................................................................................................... 46
Accessing the application .................................................................................................................................................. 47
Testing WDK samples ....................................................................................................................................................... 47

Chapter 7 Configuring User Authentication ....................................................................................................................... 49
Single Sign-On for security servers .................................................................................................................................... 49
WebSEAL Single Sign-On (SSO) authentication ................................................................................................................ 51
  Prerequisites ................................................................................................................................................................ 51
  Configurations in the wdk/app.xml file to enable WebSEAL authentication ............................................................ 51
Configuring Kerberos authentication .................................................................................................................................... 52
  Kerberos-based single sign-on authentication .............................................................................................................. 52
Table of Contents

Blank page error on deploying DA ................................................................. 75
Documentum Application Connectors refers to Microsoft Virtual
Machine installed on the client while calling UCF methods ....................... 76

Chapter 11 Deploying a Custom Application ................................................. 77
Using the comment stripper utility ............................................................. 77

Appendix A Predeployment Checklist ........................................................... 79

Appendix B Directories and Files to Back Up Before Migration ....................... 81
List of Figures

Figure 1. Basic WDK host configuration ................................................................. 14
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Preferences configuration elements</td>
<td>36</td>
</tr>
<tr>
<td>Table 2</td>
<td>Authentication elements (&lt;authentication&gt;)</td>
<td>50</td>
</tr>
<tr>
<td>Table 3</td>
<td>Authentication elements (&lt;authentication&gt;)</td>
<td>63</td>
</tr>
<tr>
<td>Table 4</td>
<td>Express user capabilities</td>
<td>69</td>
</tr>
<tr>
<td>Table 5</td>
<td>Comment stripper utility parameters</td>
<td>77</td>
</tr>
<tr>
<td>Table 6</td>
<td>Predeployment tasks</td>
<td>79</td>
</tr>
<tr>
<td>Table 7</td>
<td>Directories and files to back up</td>
<td>81</td>
</tr>
</tbody>
</table>
This guide describes how to deploy EMC Documentum Webtop and applications that are built on Web Development Kit (WDK) or Webtop.

WDK is a developer toolkit that facilitates the development of complex web-based applications connecting to EMC Documentum Content Server and content repositories. WDK contains a large library of reusable components and controls that perform common content management functions and provide a uniform user interface to applications built with WDK.

Webtop is a web application built on WDK that serves as the basis for the EMC Documentum web client applications. These applications can be customized using WDK.

Note: As newer versions of Documentum Webtop are released, all controls, actions, and components including JSP and XML that belong to Streamline view and Portlets features are deprecated from 6.6 versions of WDK-based applications. EMC does not support these features any longer. However, this document contains references and documentation on the deprecated/unsupported features for the benefit of customers that are already using these features. Please note that the entire documentation on the deprecated/unsupported features will be removed from all the Web Development Kit and Webtop product documentation as part of future Webtop releases. All the information in this document related to applets is applicable only for the Safari browser.

Intended Audience

This manual is intended primarily for administrators who are deploying an application based on WDK or Webtop. EMC Documentum web client products are built on WDK or Webtop and have their own deployment guides.

To deploy a WDK-based application, you should be familiar with the application server’s operating system and be able to install and configure a J2EE application server.

Revision History

<table>
<thead>
<tr>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2017</td>
<td>• Updated the Installing browser extension and native client application, page 20 section.</td>
</tr>
<tr>
<td></td>
<td>• Updated the Changing the classloader and compiler settings, page 45 section.</td>
</tr>
<tr>
<td>Revision Date</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>January 2017</td>
<td>• Updated the Single Sign-On for security servers, page 49 section.</td>
</tr>
<tr>
<td></td>
<td>• Updated the Enabling presets and preferences repositories, page 36 section.</td>
</tr>
<tr>
<td></td>
<td>• Added the Configuring to view preferred time zones, page 39 section.</td>
</tr>
<tr>
<td>December 2016</td>
<td>Updated the Preparing JBoss, page 26 section.</td>
</tr>
<tr>
<td>November 2016</td>
<td>Updated the Preparing Tomcat, page 29 section.</td>
</tr>
<tr>
<td>September 2016</td>
<td>Initial publication.</td>
</tr>
</tbody>
</table>
Chapter 1

Quick Start

This chapter describes the high-level steps to deploy your application.

To perform a simple product deployment

1. Plan the deployment. (Refer to Chapter 2, Planning for Deployment.)
   Check that you have required and optional supporting software, prepare the Content Server, check application server environment requirements, prepare for multiple applications, plan for language pack deployment, and (if supported) plan to deploy a customized application.

2. Prepare the clients. (Refer to Chapter 3, Preparing the Client Hosts.)
   Install a supported browser virtual machine and perform specific browser preparations for Internet Explorer, Firefox, and Chrome. If needed, you will install the mail message converter and prepare Citrix clients.

3. Prepare the application server. (Refer to Chapter 4, Preparing the Application Server Host.)
   Configure UCF, ensure you have sufficient memory allocated to the application server Java instance, turn off failover if it is not needed, and follow application-server and proxy-server specific preparation instructions.

4. Deploy the product WAR file using the application server standard deployment mechanism. (Refer to Chapter 5, Deploying a WDK-based Application.)
   You must first unpack the WAR file archive and enter some information that is specific to your environment: your connection broker and global registry information, presets and preferences repositories, and optional Federated Search server.

5. Complete the deployment. (Refer to Chapter 6, Completing the Deployment.)
   After successful deployment, you can deploy root virtual link support, enable WebSphere global security if needed, and test the application samples.
This chapter addresses software and hardware decisions you must make before you deploy a WDK-based application. This chapter contains instructions that are shared by all WDK-based products. For information on the application servers, browsers and other software in the environment that are certified for your product, see the *EMC Documentum Environment and System Requirements Guide* or the product release notes document.

Additional software products are required for WDK and WDK applications including the following:

- Content Server and its associated database
- Content Server global repository
- Connection broker
  
  You must specify one or more connection brokers in the dfc.properties file. Refer to *To configure connections in dfc.properties before deployment*, page 34 for information on configuring the connection broker before deployment.

- J2EE application server or servlet container

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

## Typical configuration

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

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

*Figure 1, page 14* shows the network components.
Caution: For security and performance reasons, do not install the Content Server and a WDK-based application on the same host. In addition, the Content Server installs an internal JBoss server that for licensing reasons cannot be used to deploy web applications.

Clustered environments — WDK-based applications can be deployed in supported clustered environments. For more information about the supported managed server configurations, see the EMC Documentum Environment and System Requirements Guide or the product release notes document.

Preparing the Content Server

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

The global registry requirement — A global registry of Content Server version 7.x must be installed in your environment in order to run a WDK-based application. For information on designating your application's global registry before deployment, refer to Enabling DFC connections to repositories, page 34.
Application server host requirements

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

Directory name restriction

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

! / : * ? " < > |

Content transfer directory permissions

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

You can change the default value to a location on your application server host for which the application server instance owner has write permission. For information on specifying locations in the UCF client and server configuration files, refer to EMC Documentum Web Development Kit Development Guide.

Some application servers require policies that grant permissions to write to these directories. The deployment information for your application server contains details on Documentum policy settings.

DNS resolution

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

Deploying multiple applications

You can deploy multiple WDK-based applications of version 7.x on a single host. Each instance of an application must be deployed to a different virtual directory. If the applications share the same application server instance, the applications must be the same version–version 7.x or higher.

You can deploy applications to separate instances of the application server. If the applications use different versions of DFC, you must deploy them in separate application server instances.
Deploying language packs

Language packs are available to localize (translate) WDK-based applications. A language pack is a language-specific archive file that contains a graphical user interface (GUI) and user documentation that have been localized into a language other than the default application language, U.S. English. To deploy language packs, unpack your product WAR file and add the language packs according to the instructions in *EMC Documentum Web Development Kit Applications Language Pack Installation and Release Notes*.

Customizing an application

A developer license is required to develop custom applications. See your EMC Documentum account representative to obtain a developer license.

**Configuration** — Configuration is defined for support purposes as changing an XML file or modifying a Java Server Page (JSP) page to configure controls on the page. Configuration does not require a developer license.

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

Customization of Documentum Administrator is not supported.
This chapter contains instructions that are shared by all WDK-based products. The *EMC Documentum Environment and System Requirements Guide* or the product release notes document contains information on the browsers that are certified for your product.

**Ensuring a certified JVM on browser clients**

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

On Windows clients, if the JVM required for UCF is not present on the client machine, UCF downloads a private JVM to the client machine. This VM does not replace the JVM that is used by the browser. For non-Windows browser hosts with a JVM of 1.7.x or later, you must pre-install a supported version of the Oracle JRE that will then be used by UCF.

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

**Supporting Outlook mail message**

New email messages through Webtop are imported in the dm_document type or any of its subtypes. Ensure that you install MailApp.dar on the Content Server to use the changed email management features.

You can view the dm_document (.msg) objects using Microsoft Outlook. To view the email messages in the .msg format, the client must have the Microsoft Outlook.

The existing EMCMF objects can be migrated using the migration utility. *EMC Documentum Email Migration User Guide* contains detailed information. If the EMCMF objects are not migrated, ES1_MRE.msi is required to view the EMCMF objects in Microsoft Outlook. To use the View as Outlook option in the HTML Viewer, the client must have the ES1_MRE.msi file. The ES1_MRE.msi file is automatically installed as part of UCF download.

To enable automatic download of the ES1_MRE.msi, uncomment the ES1_MRE.msi section in the `<WebApp Root>/wdk/contentXfer/ucf.installer.config.xml` file on the application server. The ES1_MRE.msi gets downloaded and installed on your client machine.
Preparing the Client Hosts

If ES1_MRE.msi was not already installed on the client machine, then you need to clear the existing downloaded UCF, download UCF again to install ES1_MRE.msi.

Using MailApp DAR for email messages

To enable the changed email management features, you must install MailApp.dar on the Content Server. MailApp.dar extracts the following metadata information from email messages and attaches them in the form of aspects to the email message and then imports the message into the repository as a dm_document type or any of its subtypes.

- From address
- To address
- Cc Address
- Bcc Address
- Sent date
- Received date
- Size
- Attachment count
- Message signed
- Importance
- Message Subject
- Sensitivity
- Message class
- Message id

MailApp.dar also separates the internal email attachment objects and stores them as separate objects in the repository. By default, attachment separation is disabled. This element can be enabled from the MailApp.properties file.

Configuring MailApp.properties

The MailApp.properties file is introduced for the changed email management feature.

By default, MailApp.properties is located in <WebApp Root>\WEB-INF\classes. The default values are used and it is not necessary to configure the properties. However, you can modify the value of the properties to change the default behavior.

- shouldSeparateAttachments=false
  
  If the value is false, then email message attachments are not separated.

- objectTypeForAttachments=dm_document
This indicates the object type for the attachments. It must either be dm_document or its subtypes, except dm_message_archive and its subtypes.

- **shouldParseMsgFile=true**

  If the value is true, then the .msg files are parsed. The new .msg files are parsed through MailApp.dar and attributes like From, Cc, and so on are retrieved and stored as aspect attributes. If the value is false, the .msg file are not parsed and stored as the dm_document type object without having any aspect attributes. In addition, emails search through email attributes is not possible and not displayed in the properties and listing pages.

- **shouldSkipDuplicateCheck=false**

  If the value is true, then the duplicate .msg files are imported.

### Using configuration options in app.xml for email import

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

- **<skip-duplicate-messages>:** Specifies whether to log errors for duplicate messages and continue importing (true) or throw an error and stop importing (false). The default value is true.

- **<override-object-name>:** Specifies whether you can change the object name of email objects through import. The default value is false.

### Using Citrix Presentation Server Client

Citrix Presentation Server Client can be used as a web browser.

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

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

### Ensuring active scripting is enabled

In Internet Explorer, the active scripting option is enabled by default, which is required for Webtop to work. In case active scripting is disabled, you must enable it.
Using browser extension-based content transfer mechanism

The browser extension-based content transfer mechanism supports content transfer operations in Webtop without using Java applets. This is applicable only for the Internet Explorer, Firefox, and Chrome browsers.

Content transfer mechanism consists of two components:

• content transfer browser extension: For each supported browser, there is a corresponding browser extension component. These extensions are a thin pass-through layer that allows the web page of a product to communicate with an external native application.

• native client application: This is a Java-based application and requires a supported JRE running on the client machine. The native client application is responsible for transferring content between the local file system and the application server using UCF.

Installing browser extension and native client application

Installing in Internet Explorer

Ensure the following prerequisites are met on the client machine:

• Install JRE 1.7 or later.
• Ensure that you install Microsoft .NET 4.5 or later. By default, Windows 8.1 and 10 has Microsoft .NET 4.5.
• Ensure that you have administrator privileges to install Browser Helper Object (BHO).
• If pop-up blocker is enabled, ensure that the Webtop URL is listed in the Exceptions list of pop-up blocker settings.
• Remove the TabProcGrowth entry (if available) from HKEY_CURRENT_USER or HKEY_LOCAL_MACHINE/Software/Microsoft/Internet Explorer/Main/TabProcGrowth.
• Disable the Enable Enhanced Protected Mode option in Internet Options > Advanced > Security.
• Disable the Enable 64-bit processes for Enhanced Protected Mode option in Internet Options > Advanced > Security.
• Disable the Enable Protected Mode option for trusted sites zone in Internet Options > Security.
• Add the Webtop URL to trusted sites in Internet Options > Security > Trusted sites > Sites.
• Enable the Enable third-party browser extensions option in Internet Options > Advanced > Browsing.
Preparing the Client Hosts

- Enable the **Automatically detect intranet network** option in all zones in **Internet Options > Security > Local intranet > Sites**.
- Disable the **Display intranet sites in Compatibility View** and **Use Microsoft compatibility lists** options.

Perform the following:

1. Launch Internet Explorer and log in to Webtop.
2. When prompted to install the content transfer browser extension, click **Install**.
3. Click **Run** and install BHO.
4. After the installation, click **Tools > Manage add-ons**.
5. Verify the status of **EMC Documentum Webtop Browser Helper Object**. If the status is not enabled, enable it manually.
6. Restart the browser and log in to Webtop.
7. Click **Run** and install the native client application.
8. Restart the browser and log in to Webtop.
   - You can now perform all the content transfer operations.

### Installing in Firefox

Ensure the following prerequisites are met on the client machine:

- Install JRE 1.7 or later.
- Set the Java path in **Environment Variables > System variables > Path**.
- If pop-up blocker is enabled, ensure that the Webtop URL is listed in the **Exceptions** list of pop-up blocker settings.

Perform the following:

1. Launch Firefox and log in to Webtop.
2. When prompted to install the content transfer browser extension, click **Install**.
3. When prompted that you are prevented from installing the content transfer browser extension, click **Allow** and then click **Install**.
4. Restart the browser and log in to Webtop.
5. When prompted to install the native client application, click **Save File**, click on the saved file prompt, and click **OK** once the installation is complete.
6. Restart the browser and log in to Webtop.
   - You can now perform all the content transfer operations.
Preparing the Client Hosts

Installing in Chrome

Ensure the following prerequisites are met on the client machine:

- Install JRE 1.7 or later.
- If pop-up blocker is enabled, ensure that the Webtop URL is listed in the Exceptions list of pop-up blocker settings.
- Have access to Chrome store to install the content transfer browser extension.

Perform the following:
1. Launch Chrome and log in to Webtop.
2. When prompted to install the content transfer browser extension, click Install.
3. In the EMC Documentum Client Manager dialog, click ADD TO CHROME and then click Add extension.
4. Close the dialog once you are prompted that the extension has been added.
5. Restart or refresh the browser and log in to Webtop.
6. Navigate to the downloads folder, install the native client application, and click OK once the installation is complete.
7. Restart or refresh the browser and log in to Webtop.
   You can now perform all the content transfer operations.

Note: After installing the content transfer browser extension and the native client application for all supported browsers, the UCF client will not be available on the client machine. The UCF client is downloaded only when you perform any content transfer operation for the first time. Hence, the first content transfer operation might be slow.

Locating the installation and log files

After the installation of content transfer browser extension and native client application, if no browser extension installation prompt appears in subsequent usage of Webtop application, installation is successful.
Internet Explorer, Firefox, and Chrome:

- **Installation location:** `C:\Users\<UserName>\AppData\Local\EMC\ContentXfer\com.emc.wdk.native\1`
- **Log location:** `C:\Users\<UserName>\Documentum\Logs\WDKNative`
- **(Only for Chrome) Registry entry location:** `HKEY_CURRENT_USER\SOFTWARE\Google\Chrome\NativeMessagingHosts\com.emc.wdk.native.1\`
Preparing the Application Server Host

Before you deploy a WDK-based application, ensure that your J2EE application server or Servlet container is a supported version and that it can successfully serve sample JSP pages. Your selected application server and optional external web server must be certified for the product and meet the requirements listed in the *EMC Documentum Environment and System Requirements Guide* or the product release notes document.

### Setting the Java memory allocation

The minimum recommended Oracle Java memory allocation values for application servers on a small system (only for 32-bit) is the following:

- `-Xms1024m`  
- `-Xmx1024m`

However, applying these Oracle Java memory allocation settings can slow down application servers, throw exceptions, or crash with an application that has a large number of Java Server Pages. For example, the following error is common when the value set for MaxPermSize is too low and the application server host memory is exhausted:

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

It is recommended that you set the value of the MaxPermSize parameter to 128 or higher, to avoid such errors.

Document caching can consume at least 80 MB of memory. User session caching can consume approximately 2.5 MB to 3 MB per user. Fifty connected users can consume over 200 MB of VM memory on the application server. Increase the values to meet the demands of the expected user load.

For more information on these settings, refer to Java documentation at Oracle's Java web site. More information on application server performance tuning and benchmarking for Documentum products is available from your EMC Documentum SE or EMC Documentum Consulting.

### Disabling failover

If your application server and environment combination does not support failover, you can disable failover in `app.xml` by adding the following element:

```xml
<failover>
  <enabled>false</enabled>
</failover>
```
Preparing the Application Server Host

</failover>

If you do not disable failover, you may see failover validation messages in the application server log, but these should not interfere with operations. Do not attempt to use the application in a failover environment that is not certified.

Preparing environment variables for non-default DFC locations

The base location for content transfer on the application server host is specified by the DFC environment variable dfc.data.dir. This location is specified as the value of the key dfc.data.dir in dfc.properties located within the application WAR file in WEB-INF/classes. If this variable is not set in the environment for the application server, the default location is the documentum subdirectory of the current working directory. (The current working directory contains the application server executable.) For example, in Tomcat the location is %CATALINA_HOME%/bin. On WebLogic, it is %WebLogic_HOME%/domains/wl_server/documentum.

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

Preparing JBoss

Configuring JBoss

1. If available, delete the dfc.keystore and wdk.keystore files in <JBoss Home>/bin (Windows) and <JBoss Home>/bin (Linux). This will not be present in case of a fresh installation. If present, this will be from any previous WDK application that was deployed on JBOSS.

2. To configure the dfc.properties file for the application, refer to the section Enabling DFC connections to repositories, page 34.

3. To configure encrypted passwords in the app.xml file using TrustedAuthenticatorTool, refer to the section Configuring encrypted password for presets and preferences repositories, page 37.

4. Encrypting the password using TrustedAuthenticatorTool creates the dfc.keystore and wdk.keystore in the WEB-INF/classes folder.

5. Move the keystore files from <WebApp Root>/WEB-INF/classes (Windows) and <WebApp Root>/WEB-INF/classes (Linux) to the bin folder of the <JBoss Home> directory.

6. Copy the contents of the classes folder from <WebApp Root>/WEB-INF/classes (Windows) and <WebApp Root>/WEB-INF/classes (Linux) to a temporary location (for example, Temp-Loc).

   Execute the following command at Temp-Loc to create a web-inf-classes jar file:
7. Copy the `web-inf` folder to `<WebApp Root>`\WEB-INF\lib (Windows) and `<WebApp Root>/WEB-INF/lib (Linux).

8. Delete the `classes` folder from `<WebApp Root>`\WEB-INF (Windows) and `<WebApp Root>/WEB-INF (Linux).

9. Add the `shared` entry (in bold) to the `subsystem` tag in the `standalone.xml` file in `<JBOSS Home>`\standalone\configuration (Windows) and `<JBOSS Home>`\standalone\configuration (Linux) to disable shared pooling:

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

10. Configure the binding address by replacing 127.0.0.1 with the application server host IP address in `<wsdl-host>` and `<interfaces>` tags in `standalone.xml`.

11. Execute the following command at `<WebApp Root>` to repackage the Webtop WAR file:

    ```
jar -cvf webtop.war *
    ```

## Deploying multiple applications on JBoss

JBoss requires the DFC and WDK keystores in the JBOSS/bin folder. If multiple applications with different preset or preference repository passwords are deployed, then the DFC and WDK keystore files in the JBOSS/bin folder should have the encryption keys to decrypt both the encrypted passwords present in the `app.xml` files of both the applications.

1. Create an XML file with the name `jboss-deployment-structure.xml` and add the following tags to the file:

   ```xml
   <jboss-deployment-structure>
   <deployment>
   <exclusions>
   <module name="org.apache.log4j"/>
   </exclusions>
   </deployment>
   <jboss-deployment-structure>
   ```

2. Add the `jboss-deployment-structure.xml` file in the `WEB-INF` folder.

3. To configure the `dfc.properties` file for the application, refer to the section Enabling DFC connections to repositories, page 34.

4. To generate the keystores for both the applications, perform either of the following options:
Preparing the Application Server Host

Option 1
1. For application 1, configure encrypted passwords in the app.xml file using TrustedAuthenticatorTool. For more information, refer to the section Configuring encrypted password for presets and preferences repositories, page 37.

2. Encrypting the password using TrustedAuthenticatorTool creates the dfc.keystore and wdk.keystore files in the WEB-INF/classes folder.

3. Copy the DFC and WDK keystores from application 1 to the application 2 (classes folder) and encrypt the preference repository password of application 2 using TrustedAuthenticatorTool. For more information, see Configuring encrypted password for presets and preferences repositories, page 37.
   This updates the same keystore file with the encryption keys to decrypt the password for the second repository as well.

4. Move the updated keystore files from application 2 to the JBOSS/bin folder.

Option 2
1. Encrypt the preference repository passwords for multiple applications in the same location. For example, navigate to the <WebApp Root>/WEB-INF/classes folder of application 1 and encrypt the preference repository passwords for both the applications. The app.xml files of both the applications are updated with the respective encrypted password generated for the global repository mentioned in the dfc.properties file of the application. For more information, refer to the section Configuring encrypted password for presets and preferences repositories, page 37.

2. Move the keystore file which has both the encryption keys from <WebApp Root>/WEB-INF/classes (Windows) and <WebAppRoot>/WEB-INF/classes (Linux) to the bin folder of the <Jboss Home> directory.

5. For application 1 and application 2, copy the contents of the classes folder from \WEB-INF\classes (Windows) and /WEB-INF/classes (Linux) to temporary locations. For example, Temp-Loc1 and Temp-Loc2.
   Execute the following command at Temp-Loc1 and Temp-Loc2 to create a web-inf-classes.jar files for the respective applications file:
   jar -cvf web-inf-classes.jar *

6. For application1 and application 2, copy the respective web-inf-classes.jar file to <WebApp Root>/WEB-INF\lib (Windows) and <WebApp Root>/WEB-INF/lib (Linux) folder structure.

7. For application1 and application 2, delete the corresponding classes folder from <WebApp Root>/WEB-INF (Windows) and <WebApp Root>/WEB-INF (Linux) folder structure.

8. If you are configuring the JBOSS application server for the first time, add the configuration entry (in bold) to the subsystem tag in the standalone.xml file and configure the binding address as mentioned in the steps 9 and 10 of Preparing JBoss, page 26 section.

9. For both the applications execute the following command at <WebApp Root> to repackage the Webtop WAR file:
   jar -cvf webtop.war *
Enabling HTTPOnly Cookies Support

For the HTTPOnly cookies support, navigate to `WEB-INF/web.xml` and perform the following:

1. Update the web-app header specification from version 2.4 to 3.0:

   From
   
   ```xml
   <web-app version="2.4" xmlns=http://java.sun.com/xml/ns/j2ee
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
   http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">``

   To
   
   ```xml
   <web-app version="3.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
   metadata-complete="true"
   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
   http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"/>
   ```

2. Add the following entry in `<session-config>`:

   ```xml
   <cookie-config>
   <http-only>true</http-only/>
   </cookie-config>
   ```

Preparing Tomcat

You must disable tag reuse in `/conf/web.xml`. Find the JSP servlet entry and add the following:

```xml
<init-param>
<param-name>enablePooling</param-name>
<param-value>false</param-value>
</init-param>
```

For the HTTPOnly cookies support, perform the following:

- Add the following in `server.xml` within the `<Engine>` and `<Host>` tags:

  ```xml
  <Context path="/webtop" useHttpOnly="true"/>
  ```

If there are multiple applications deployed, then the HTTPOnly property can be set for each application in Tomcat's `server.xml`. For example:

```xml
<Context path="/da" useHttpOnly="false"/>
<Context path="/webtop" useHttpOnly="true"/>
<Context path="/dam" useHttpOnly="false"/>
```

Preparing vFabric tc Server

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

For the HTTPOnly cookies support, perform the settings as described in Preparing Tomcat, page 29.
Preparing WebLogic server

The Preparing the WAR file for deployment, page 33 section contains detailed information on deploying the WAR file.

Supporting large content transfer operations in a managed server environment

If you are deploying in a WebLogic Managed Server environment and you use UCF to perform large content operations, set the WLIOTimeoutSecs parameter for the web server plugin to a very large value. UCF requires a sticky session for a single operation. The Oracle documentation contains additional details on Web Server Plug-in parameters.

Preparing IBM WebSphere

The following section describes how to prepare the application server to support failover in a cluster, and to support non-default content transfer locations. The Preparing the WAR file for deployment, page 33 section contains detailed information on deploying the WAR file.

For the HttpOnly cookies support, enable the HttpOnly property by selecting Set session cookies to HTTPOnly to help prevent cross-site scripting attacks from the Application servers > server1 > Session management > Cookies location.

If there are multiple applications deployed in the same application server and if you require to set the HttpOnly property just for WDK application, then perform the following:

1. Select Set session cookies to HTTPOnly to help prevent cross-site scripting attacks from All Applications > webtop > Session management > Cookies.

2. Select Override the session management from All Applications > webtop > Session management > Cookies.

3. Select a distributed session from Distributed sessions same as servers (for example, if you had selected Memory-to-memory replication on your server then you must select Memory-to-memory replication) from All Applications > webtop > Session management > Distributed environment settings.

Preparing the Application Server for Java 2 security

If you plan to use Java 2 security for securing access to available system resources in your Webtop installation, then you must use the Java policy configuration file that is bundled with your application server. The Java policy configuration file of the application server specifies the permissions granted to the classes, in your Webtop installation. To help you update the Java policy configuration file of the application server, an example policy template file is included in the Webtop installation.
Preparing the Application Server Host

(Webtop.example.java.policy file). The file specifies the permissions required to access the Webtop classes. The Webtop.example.java.policy file is included in the webtop.war file, and gets extracted into the <webtop_app_root> folder.

The Webtop.example.java.policy file contains a default set of permissions that are required for Webtop functionality. If you customize Webtop, then you must incorporate other relevant permissions in your application server policy file. EMC Documentum does not support such changes to the application server policy files.

To enable Java 2 security in the application server:

1. Navigate to webtop_app_root\Webtop.example.java.policy and identify the permissions that must be incorporated into the application server security policy file.

2. Navigate to the policy file of your application server.

   Based on the syntax and locations specified in the application server documentation, add, or update the permissions (identified in the Webtop.example.java.policy file), in the policy file of the application server.

   ! Caution: Do not omit any permission specified in the Webtop.example.java.policy file while incorporating the permissions in the application server Java policy configuration file. Otherwise, Webtop may fail to start or some features might fail to work.

3. Configure your application server to pick the security policy files.

Preparing to use an external web server

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

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

If you are deploying in a manager server or network deployment environment, the external web server must provide session affinity support.
Chapter 5

Deploying a WDK-based Application

This chapter contains instructions that are shared by all WDK-based products.
After you complete the required predeployment tasks, deploy a WDK application on the application server host.

Preparing the WAR file for deployment

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

To deploy a WDK-based application:

1. Download the WDK application WAR file to a temporary directory on the application server host.
2. Unpack the WAR file and modify the dfc.properties file following the instructions in Enabling DFC connections to repositories, page 34. You must perform this procedure before attempting to connect to Documentum repositories.
3. Enable the presets and preferences repositories in app.xml following the instructions in Enabling presets and preferences repositories, page 36.
4. Add or migrate customizations from previous WDK-based applications.
5. Apply language packs if you have purchased them.
6. Make any UCF configuration changes that your applications needs before deploying. The EMC Documentum Web Development Kit Development Guide contains more details.
7. If deploying to a production environment, remove the following files:
   - <WebApp Root>\unstripped.jar
   - <WebApp Root>\webtop.example.java.policy
   - <WebApp Root>\webtop\webtop.vep
   - <WebApp Root>\webtop\webtop BEA.vep
   - <WebApp Root>\webtop\src
8. Re-archive the WAR file.
9. Deploy the WAR file according to the deployment instructions in your application server documentation.
Enabling DFC connections to repositories

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

A global registry of Content Server version 7.x is required for WDK-based applications. The global registry is a central repository that serves several purposes:

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

The EMC Documentum Content Server Installation Guide contains information about enabling a repository as a global registry.

**To provide connection broker and global registry values of the content server, in the dfc.properties file of custom WDK Application WAR file:**

Configure the dfc.properties file of the application to connect the application to the content server and the repository.

1. On the global registry repository host, locate the Content Server installation directory. On Windows hosts, the default installation directory is C:\Documentum. On UNIX hosts, this directory is specified by the environment variable $DOCUMENTUM.
2. Navigate to the config subdirectory and open the dfc.properties file.
3. Copy the following keys and their values from the file:
   - dfc.docbroker.host[0]=address
   - dfc.globalregistry.repository=repository_name
   - dfc.globalregistry.username=username
   - dfc.globalregistry.password=encrypted_password
   - dfc.docbroker.port[0]=port_number
4. Unpack the custom WDK Application WAR file to the ROOT directory of the application server.
5. Open the dfc.properties file located in WEB-INF/classes within this expanded WAR file directory.
6. Paste in the values that you copied from the global registry dfc.properties.
7. Use a text editor to configure additional properties in this file or make any changes to it.
8. Save the dfc.properties file and deploy the application.

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

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

1. Unpack the application WAR file.
2. Open the file dfc.properties in WEB-INF/classes.
3. Add the fully qualified hostname for the docbroker to the following key. You can add backup hosts by incrementing the index number within brackets.
dfc.docbroker.host[0]=host_name

4. If you wish to use a port for the docbroker other than the default of 1489, add a port key to dfc.properties:
   dfc.docbroker.port=port_number

5. Add the global registry repository name to the following key:
   dfc.globalregistry.repository=repository_name

6. Add the username of the dm_bof_registry user to the following key:
   dfc.globalregistry.username=dm_bof_registry_user_name
   The global registry user, who has the username of dm_bof_registry, has read access to objects in the /System/Modules and /System/NetworkLocations only.

7. Add an encrypted password value for the following key:
   dfc.globalregistry.password=encrypted_password
   You can either copy the username and encrypted password from the dfc.properties file on the global registry Content Server host, or you can select another global registry user and encrypt the password using the following command from a command prompt (assumes the directory containing javaw.exe is on the system path):
   java -cp dfc.jar com.documentum.fc.tools.RegistryPasswordUtils
   password_to_be_encrypted

Enabling DFC memory optimization

You can free up memory resources by adding the following line to your dfc.properties file to define and set DFC diagnostics:
   dfc.diagnostics.resources.enable=false

If you want to identify the code that causes the session leaks problem, set the value of the dfc.diagnostics.resources.enable parameter in dfc.properties to true.

For instructions about unpacking the war file of your application and modifying dfc.properties, see Enabling DFC connections to repositories, page 34.

Configuring UCF

The EMC Documentum Web Development Kit Development Guide contains the following procedures:
- How to configure different content transfer mechanisms (UCF or HTTP) for roles.
- How to configure the UCF client content transfer directories, including client path substitution.
- How to support self-signed or unsigned SSL certificates.
- How to configure the UCF server for forward and reverse proxy servers and alternative chunking.

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

## Forcing UCF to install a configured JRE

If your WDK-based application uses UCF content transfer, it is mandatory that the browser has a Oracle JRE installed. By default, the UCF installer uses the Oracle JRE that is installed in the browser if its version is the same as or later than the version of JRE in the UCF installer. A later version of JRE sometimes introduces problems in an application.

If you do not want to allow multiple JRE versions, you can configure the UCF installer to use or install only the version that is configured in the installer configuration file. You must add an enforceJreInstallation attribute to the runtime Java element in the file ucf.installer.config.xml to use the configured JRE version. This file is located in your web application directory, wdk/contentXfer. Change the runtime Java element by adding the enforceJreInstallation attribute.

```xml
platform os="windows" arch="x86">
  <runtime type="java" version="<JRE Version>" href="<JRE Bundled Zip File>"
    exePath="<JRE Path>\bin\java.exe" enforceJreInstallation="true">
    </runtime>
</platform>
```

For example:

```xml
platform os="windows" arch="x86">
  <runtime type="java" version="1.7.0.91" href="win-jrel.7.0_91.zip"
    exePath="jrel.7.0_91\bin\java.exe" enforceJreInstallation="true">
    </runtime>
</platform>
```

If users have already installed UCF, you must force an update of the UCF configuration every time you make changes to the UCF configuration on the application server. Ensure that you append a new letter to the app version to force the update. In the following example, the version 6.7.0.223 is changed:

```xml
<app id="shared" version="6.7.0.223a" compatibilityVersion="5.3.0"/>
```

## Enabling presets and preferences repositories

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

Add your preferences repository settings to app.xml in the /custom directory of the application. Copy the entire `<preferencesrepository>` element from /wdk/app.xml into /custom/app.xml and then specify your repository. The **EMC Documentum Web Development Kit Development Guide** contains detailed information on other preferences settings in app.xml.

### Table 1. Preferences configuration elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;preferencesrepository&gt;</code></td>
<td>Contains a <code>&lt;repository&gt;</code> element. If this element is not present, user preferences are stored in the global repository, which can slow down performance.</td>
</tr>
</tbody>
</table>
Deploying a WDK-based Application

Element | Description
--- | ---
`.repository_path` | Specifies the path within the preference repository in which to store preferences. If the path does not exist at application startup, it will be created.

`.repository` | Specifies the repository in which to store preferences, preferably not the global repository.

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

### Configuring encrypted password for presets and preferences repositories

1. Change the default passwords in Content Server.
   
   For both `dmc_wdk_presets_owner` and `dmc_wdk_preferences_owner` users, the default password must be changed using IAPI in Content Server. To do this, login to IAPI as an Administrator and execute the following commands.
   
   - To change the password for the `dmc_wdk_presets_owner` user:
     ```
     retrieve,c, dm_user where user_name='dmc_wdk_presets_owner';
     set,c,l, user_password
     <enter new password>
     save,c,l
     ```
   
   - To change the password for the `dmc_wdk_preferences_owner` user:
     ```
     retrieve,c, dm_user where user_name='dmc_wdk_preferences_owner';
     set,c,l, user_password
     <enter new password>
     save,c,l
     ```

2. Encrypt the passwords in Webtop using the TrustedAuthenticatorTool available in the WEB-INF/classes folder.

   **Windows:**
   
   To create an encrypted password, execute the following command at the command prompt:
   ```
   java TrustedAuthenticatorTool <password>
   ```
   
   The utility sends the encrypted password to the standard output.

   **Oracle Java example:**
   ```
   C:\webtop\WEB-INF\classes>java -cp .;./lib/dfc.jar;./lib/commons-io-1.2.jar; ./lib/certjFIPS.jar;./lib/jsafeFIPS.jar TrustedAuthenticatorTool trusted
   Encrypted: [5P54fOKuCKM=], Decrypted: [trusted]
   ```

   **IBM Java example:**
   ```
   C:\IBM\WebSphere\AppServer\java\bin\java.exe -cp .;./lib/dfc.jar; ./lib/commons-io-1.2.jar;./lib/certjFIPS.jar;./lib/jsafeFIPS.jar TrustedAuthenticatorTool trusted
   Encrypted: [5P54fOKuCKM=], Decrypted: [trusted]
   ```
Deploying a WDK-based Application

Linux:
1. Navigate to the WEB-INF/classes folder.
2. Set the classpath for the referenced jars:
   ```bash
   export JAR_PATH=..:/lib/dfc.jar:..:/lib/commons-io-1.2.jar:..:/lib/certJFIPS.jar:..:/lib/jsafeFIPS.jar
   ```
3. Execute the Java command to generate the encrypted password:
   ```bash
   java -cp $JAR_PATH TrustedAuthenticatorTool trusted
   ```
4. Update the encrypted passwords in app.xml for Webtop.
   Search for `<presets>` and update the `<password>` attribute with the encrypted password. For example:
   ```xml
   <presets>
   ...<password>5P54fOKuCKM=</password>
   ...</presets>
   ```
   Search for `<preferencesrepository>` and update the `<password>` attribute with the encrypted password. For example:
   ```xml
   <preferencesrepository>
   ...<password>5P54fOKuCKM=</password>
   ...</preferencesrepository>
   ```
4. Delete the Documentum folder in `<WebApp Root>`\WEB-INF\classes (Windows) and `<WebApp Root>/WEB-INF/classes (Linux).
5. Start the application server.

Enabling HttpOnly cookies support for Webtop

To enable the HttpOnly cookies support for Webtop, the following is added to wdk/app.xml:
```xml
<httponly_cookies>
<enabled>true</enabled>
</httponly_cookies>
```

By default, the value of the property is set to true. `<httponly_cookies>` property is only for Webtop cookies. JSESSIONID cookie can be marked as HttpOnly through respective application server settings.

Configuring to validate application URL

The following configuration has been added to enable or disable the validation of application URLs in `<WebApp Root>`\wdk\app.xml:
```xml
-uri-validation>
<enabled>true</enabled>
</uri-validation>
```
By default, `<enabled>` is set to `true` to validate the application URLs.

**Note:**
- EMC recommends that administrators set `<enabled>` to `true`.
- When you set `<enabled>` to `false` and if the customized application URLs work, then the customized application URLs may be affected and needs to be fixed.

## Configuring for improved advanced search

The flag, `data_handler_fetch_by_range`, is added in `<WebApp Root>/wdk/app.xml` and is used for an improved advanced search capabilities.

This flag enables objectlist cache to be efficiently filled up using range based DQL query. When the user clicks next page, only the results for the next page will be fetched from the Content Server. By default, the value is set to `true`. For example, if the user clicks next page at page 5 and the items per page is 100, then Webtop fetches the results from 501 to 600.

To fetch all the results at once, set the value to `false`.

## Configuring search results

The flag, `fetchresultsbypage`, is added in `<WebApp Root>/webcomponent/config/library/search/search/ex/search60_component.xml` to get the search results per page. You must set the value to `false` to get the complete search results. By default, the value is set to `true`.

## Configuring to view preferred time zones

You can configure the following time zone configurations in `wdk/app.xml` to view the time zone of Content Server, application server, or the client in the Webtop application:

- To view the time zone at the Content Server location, set the `r_tz_aware` attribute of the `dm_docbase_config` Content Server object to `False`, and set the `client_timezone_awarness` attribute to `False` in `wdk/app.xml` on the application server machine
- To view the time zone at the application server location, set the `r_tz_aware` attribute of the `dm_docbase_config` Content Server object to `True`, and set the `client_timezone_awarness` attribute to `False` in `wdk/app.xml` on the application server machine
- To view the time zone at the client machine location, set the `r_tz_aware` attribute of the `dm_docbase_config` Content Server object to `True`, and set the `client_timezone_awarness` attribute to `True` in `wdk/app.xml` on the application server machine

After you make the specified changes, import or checkout any file from Webtop to verify if the time zone settings work properly.
Enabling retention of folder structure and objects on export

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

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

By default, the value is set to false.

Enabling external searches

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

Configuring the connection to the search server

The following procedure describes how to enable the Federated Search server to query external sources.

To configure the connection to a Federated Search server:

1. Unpack the client application WAR file.
2. Open the file dfc.properties in WEB-INF/classes.
3. Enable the Federated Search server by setting the following:
   ```
dfc.search.ecis.enable=true
   ```
4. Specify the RMI Registry host for the Federated Search server by setting the following:
   ```
dfc.search.ecis.host=host_IP
dfc.search.ecis.port=port
   ```
   Where
   - `host_IP` is IP address or machine name of the Federated Search server.
   - `port` is the port number that accesses the Federated Search server. The default port is 3005.

Configuring the connection to the backup search server

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

- \texttt{dfc.search.ecis.backup.host}: host of the backup Federated Search server. Default value is: localhost.
- \texttt{dfc.search.ecis.backup.port}: port of the backup Federated Search server. Default value is: 3005.
- \texttt{dfc.search.ecis.retry.period}: waiting period before retrying to connect to the primary Federated Search server. This time is in milliseconds. Default value is: 300000.

### Enabling xPlore search with Webtop emails

To enable the xPlore search for emails imported using MailApp.dar, perform the following:

1. Run the following query in the repository:
   
   ```
   ALTER ASPECT mdmo_message_aspect FULLTEXT SUPPORT ADD ALL
   ```
2. Stop the xPlore service in the index machine.
3. Clear the BOF cache at `<xPlore Home>
   `<JBoss version>
   `<server>DctmServer
   `Indexagent\data\Indexagent\cache\content_server_version\bof\repository_name`.
4. Start the xPlore service in the index machine.

### Deploying multiple applications

Two or more WDK-based applications of version 6.x or higher can share the same application server instance if they are version 6 or higher.

The EMC Documentum Web Development Kit Development Guide contains detailed information on configuring custom HTTP session attributes and configuring URL addressable components and actions.

### Enabling Documentum Collaborative Services

Web Development Kit (WDK) 6.5 SP2 and later versions support enabling and disabling of Documentum Collaborative Services (DCS) in Webtop. DCS can be enabled or disabled in Webtop by modifying the Custom\app.xml of your WDK application with the configuration described in this section. By default, the DCS feature is disabled in the Webtop application.

**Note:** You must ensure that the DCS DAR file has been installed on the repository before you enable the DCS functionality for Web Development Kit (WDK) 6.5 SP2 and later version applications.
To ensure that the DCS DARs have been installed:

1. Log in to Webtop.
2. Select File > New. The Calendar and Data Table options are enabled in the menu if the DCS DARs have been installed.

The EMC Documentum Composer User Guide contains more information.

To enable Documentum Collaborative Services:

1. Open the app.xml file in the /wdk folder of your application.
2. Change the value of the <enabled> tag to "true" in the following <config> tag:

   <config>
   
   <scope>
   
   <application>
   
   <collaboration-support>
   
   <enabled>true</enabled>
   
   </collaboration-support>
   
   </application>
   
   </scope>
   
   </config>

3. Save and close the app.xml file.

To enable Documentum Collaboration Services using Documentum Administrator:

1. Connect to Documentum Administrator.
4. Click Enable.
5. Specify the Documentum Collaboration Service license key as follows: JDASPCAKSDE

Note:

- Note that the performance of Webtop may be impacted marginally after enabling DCS because Webtop will include the extra functionality.
- Webtop 6.5 SP2 and later versions customers: Download DCS-related dar file(s) and install the DCS DAR file(s) on the repository. The collaboration features are turned on automatically on the Content Server. No license key is required to enable DCS.
- You should install the DCS 6.5 or later version DAR file because of fixes to several critical issues.

Configuring Client capability roles

WDK-based custom applications can be configured to use any role that is defined in the repository. If roles are not configured for users in the repository, WDK defaults to using the client capability model with the following client_capability attributes that can be set on the dm_user object: consumer, contributor, coordinator, and administrator. Client capabilities refer to the operations you can execute on an object of a WDK-based custom application depending on your role.

If the client capability level is not set for a user, the role service assigns the user the consumer role.
A user can perform all operations on an object if the user owns the object, regardless of the user’s role. See the *EMC Documentum Web Development Kit Development Guide* for more information about the actions that can be executed against an object depending on the user’s role and configuring client capability roles.

**Deploying Documentum Webtop in a cluster for Load Balancing and Failover scenario**

In a load balancing and failover scenario, you must set up a cluster environment using a combination of the Oracle WebLogic Configuration tool and Oracle WebLogic application server, or IBM WebSphere Network Deployment Manager and IBM WebSphere application server. The *EMC Documentum Webtop Release Notes* contains the information.
Chapter 6

Completing the Deployment

This chapter contains instructions that are shared by all WDK-based products. After you deploy a WDK application, there are additional procedures that you may need to perform in order to finish and verify the deployment.

Configuring IBM WebSphere after deployment

To complete the deployment, perform the following procedures.

Changing the classloader and compiler settings

Change the classloader setting for the WDK-based application module in WebSphere, in the Manage Modules section of the administration console. Select the WAR file and for Classloader order choose Classes loaded with local class loader first (parent last), then click Save.

Precompile JavaServer pages:

- While deploying the application, you can select Precompile JavaServer Pages files configuration in the Deployment Manager Administrator console.

- After the application is deployed, you can edit the JSP engine configuration parameters in the WEB-INF/ibm-web-ext.xmi file to change its JSP compiler options. Set the JSP compiler option to useJDKCompiler to true and the source level to 1.7 (JRE 7) in the configuration file ibm-web-ext.xmi under the application deployment directory. For example:

  ```xml
  <WAS_HOME>/AppServer/profiles/AppSrv01/config/cells/<cell_name>/
  applications/webtop_war.ear/deployments/webtop_war/webtop.war/
  WEB-INF/ibm-web-ext.xmi
  
  Configure the settings as follows:
  
  <jspAttributes xml:id="JSPAttribute_1486449242438"
  name="jdkSourceLevel" value="17"/>
  ```
Setting custom webcontainer property com.ibm.ws.webcontainer.invokefilterscompatibility to true

If you are using IBM WebSphere, you must add the webcontainer com.ibm.ws.webcontainer.invokefilterscompatibility custom property and set it to true using the WebSphere Admin console; otherwise, all UCF content transfer operation will fail. For more information about setting webcontainer custom properties, see Setting webcontainer custom properties.

Deploying default virtual link support

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

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

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

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

To install default virtual link support:

1. Deploy the vlink.war file as the root web application on the application server.
   Some application servers have an existing root web application which you must replace with the default virtual link application. Others require you to create a root web application manually or during application server installation. Refer to the documentation for the application server for information on a root web application.

2. Deploy the virtual link war file (vlink.war or ROOT.war) to the application server by using the mechanism recommended by the application server for deploying a default web application.

3. Modify the DefaultWdkAppName param-value in the web.xml of the virtual link WAR file. This parameter value specifies the WDK-based application that will handle the virtual link request if there is no current repository session for the user. If you do not specify a parameter value, it will default to webtop.

   On Weblogic, add the following line to weblogic.xml file or use the weblogic.xml file that is bundled with vlink.war:
   
   `<context-root>/*</context-root>`

The EMC Documentum Web Development Kit Development Guide contains more information on virtual links.
Accessing the application

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

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

To verify the deployment and configuration of a WDK application:

1. Open a browser window and type this URL:
   
   http://host_name:port_number/virtual_directory

   Where:
   
   • host_name is the host where the application server is installed. If the browser is on the application server machine, substitute localhost for host_name; for example, http://localhost:8080/webtop.
   
   • port_number is the port where the application server listens for connections
   
   • virtual_directory is the virtual directory for your application

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

2. Log in to a repository through the WDK-based application.
   
   If the login succeeds, the application is correctly deployed and configured.

Testing WDK samples

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

To view the WDK samples:

1. Ensure that the application server is running.

2. Open a browser and type the following URL:
   
   http://host_name:port_number/virtual_directory/component/login

3. Log in to a test repository.

4. Download the WDK_Samples_and_TestBed zip file from EMC Online Support (https://support.emc.com). Copy the samples folder and paste it in the WDK folder of the deployed Webtop application.
   
   Type this URL:
   
   http://host_name:port_number/virtual_dir/wdk/samples/index.jsp

   This page displays a list of the available samples.
5. Click **Session Zoo**, type the relevant credentials, and click **Create Connection**.
6. Continue to experiment with other samples, especially Menu Zoo, Tree Control, and FX Control Pens. Some samples have **Create Test Cab** and **Destroy Test Cab** buttons. These create and delete a test cabinet in the repository and require Create Cabinet privileges.
Chapter 7

Configuring User Authentication

This chapter describes how to configure user authentication.

Single Sign-On for security servers

WDK applications support SSO using RSA Access Manager (formerly known as ClearTrust).

RSA Access Manager users must have the same login names as the Content Server repository. User names are case-sensitive for the Content Server, so Access Manager user names must be a string with any value between 1 and 255 characters in length and have the same case as the repository login. Errors in authentication are logged in the /Documentum/dba/log/dm_rsa.log file.

To enable single sign-on (SSO):

1. Configure the RSA Access Manager security server to authenticate repository users. (For more information, see the security server documentation.)

2. Configure the web application server to use an external HTTP Server supported by the security server. (For more information, see the RSA security server documentation.)

3. Configure the Content Server plugin. (For more information, see the EMC Documentum Content Server Administration and Configuration Guide.)

4. Configure the WDK-based application in app.xml as described in To configure app.xml for a security server single sign-on:, page 50.

5. Create a directory named rsaConfig under the root WDK-based application directory. Copy two files: aserver.conf from the Access Manager server and webagent.conf from the RSA web agent. Paste them into the rsaConfig directory.

   If you make changes to the original files, you must copy the changed files to your WDK-based application rsaConfig directory. For more information on these files, refer to the RSA documentation.

6. Locate the file AuthenticationScheme.properties in WEB-INF/classes/com/documentum/web/formext/session. The SSO authentication scheme classes. Modify the properties file to make your preferred SSO authentication scheme (SSOAuthenticationScheme or RSASSOAuthenticationScheme) first in the list of authentications that are attempted during login.

   If the Docbase Login scheme is listed before the SSO scheme, the user is presented with a login screen instead of single sign-on.
7. Restart the application server.

**To configure app.xml for a security server single sign-on:**

1. Open the app.xml file in your applications/custom directory.
2. Copy from app.xml the <authentication> element and its entire contents, and paste into your custom app.xml.
3. Update the <sso_config> element under the existing <authentication> element as shown in the following example:

   ```xml
   <authentication>
     <domain/>
     <docbase>secure_docbase</docbase>
     <service_class>
       com.documentum.web.formext.session.AuthenticationService
     </service_class>
     <sso_config>
       <ecs_plug_in>dm_rsa</ecs_plug_in>
       <ticket_cookie>CTSESSION</ticket_cookie>
       <user_header>HTTP_CT_REMOTE_USER</user_header>
     </sso_config>
   </authentication>
   ```

Table 2, page 50 describes valid values for each element.

**Table 2. Authentication elements (<authentication>)**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;docbase&gt;</td>
<td>Specifies default repository name. When SSO authentication is enabled but a repository name is not explicitly spelled out by the user nor defined in this element, the sso_login component is called. In this case the component prompts the user for the repository name.</td>
</tr>
<tr>
<td>&lt;domain&gt;</td>
<td>Specifies Windows network domain name.</td>
</tr>
<tr>
<td>&lt;service_class&gt;</td>
<td>Specifies fully qualified name of class that provides authentication service. This class can perform pre- or post-processing of authentication.</td>
</tr>
<tr>
<td>&lt;sso_config&gt;</td>
<td>Contains SSO authentication configuration elements.</td>
</tr>
<tr>
<td>&lt;sso_config&gt;</td>
<td>Specifies name of the Content Server authentication plugin (not the authentication scheme name). Valid value: dm_rsa</td>
</tr>
</tbody>
</table>
### WebSEAL Single Sign-On (SSO) authentication

EMC Documentum can integrate with WebSEAL, its SSO solution, or any other SSO solution supported by WebSEAL.

For more information about installing and configuring the WebSEAL server, see the related IBM documentation. For more information about configuring Documentum applications to enable WebSEAL SSO authentication, see the Development Guide or Installation Guide of your application.

For the **HttpOnly** cookies support, set the value of the `pass-http-only-cookie-attr` property in `webseald-default.conf` to `yes`. This allows WebSEAL to pass the **HttpOnly** property from junction set-cookie headers through to clients. By default, the value is set to `no` to discard the **HttpOnly** property from the cookie.

### Prerequisites


- Install the IBM WebSEAL server on a machine, and create a HTTP or HTTPS junction that will link the WebSEAL server to Webtop.

- Deploy Webtop on the application server machine, and connect to a Content Server that has been configured for WebSEAL SSO authentication. See Chapter 5, *Deploying a WDK-based Application* for more information to deploy Webtop on an application server. For more information about configuring Content Server for WebSEAL SSO authentication, see the *EMC Documentum Content Server Installation Guide*, *EMC Documentum Content Server Administration and Configuration Guide*, and the *EMC Documentum Administrator User Guide*.

### Configurations in the wdk/app.xml file to enable WebSEAL authentication

Set the value of the `user_header` tag to `iv-user`, within the authentication tag:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;sso_config&gt;</code></td>
<td>Specifies name of vendor-specific cookie that holds the sign-on ticket.</td>
</tr>
<tr>
<td><code>&lt;ticket_cookie&gt;</code></td>
<td>Valid value: CTSESSION</td>
</tr>
<tr>
<td><code>&lt;sso_config&gt;</code></td>
<td>Specifies name of vendor-specific header that holds the username.</td>
</tr>
<tr>
<td><code>&lt;user_header&gt;</code></td>
<td>Valid value: HTTP_CT_REMOTE_USER.</td>
</tr>
</tbody>
</table>
<authentication>
  <webseal_config>
    <user_header>iv-user</user_header>
  </webseal_config>
</authentication>

Note: Copy the user_header element into the authentication tag of the custom/app.xml file.

### Configuring Kerberos authentication

Kerberos SSO authentication scheme is used to authenticate the user who wants to log in to the WDK application from a computer that is in the Kerberos domain. When a user accesses the WDK application's URL for the first time, the user is prompted to select a repository and log in. Upon subsequent connections to the same repository (unless the browser cache is cleared), the user is automatically logged into the repository.

### Kerberos-based single sign-on authentication

When Kerberos-based Single Sign-On Authentication is enabled on a WDK application, users are automatically authenticated and logged in to the repository using their credentials stored in the user's private credential area on the Windows platform.

### Prerequisites

- Deploy the WDK application on the application server machine, and connect to a Content Server that has been configured for Kerberos SSO authentication. For more information about configuring Content Server for Kerberos SSO authentication, see the *EMC Documentum Content Server Administration and Configuration Guide*.
- Install a supported browser on the client machine.
- Register the WDK application as a Service Principal in the Key Distribution Center (KDC). Refer to the *Create user account for the WDK application in the active directory*, page 54 section for more information on registering the WDK application as a Service principal in the KDC.
- On a Windows Server 2008 R2 SP1 x64 Edition host, to make the TGT's session key available for Java to use to acquire additional service tickets, make sure that the following key and value have been added to the registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
Value Name: allowtgtsessionkey
Value Type: REG_DWORD
Value: 0x01
```

Note: By default, Windows does not allow the TGT's session key to be accessed.
Configuring wdk/app.xml to enable Kerberos authentication

Carry out the configurations specified in this section, in the <enabled>, and <domain> tags within the <authentication> tag, and copy the configurations into the custom/app.xml file.

Enabling Kerberos authentication in the WDK application

An application level setting is provided in wdk/app.xml within the <authentication> tag to enable or disable Kerberos-based SSO authentication. The default value defined for the <enabled> tag in the <kerberos_sso> element is false. Set the <enabled> tag to true to enable Kerberos SSO authentication.

```
<kerberos_sso>
  <enabled>true</enabled>
</kerberos_sso>
```

Configuring the Kerberos domain name

An application level tag is provided to specify the Kerberos domain, within the <authentication> tag. Enter the domain name in the <domain> tag.

```
<kerberos_sso>
  <domain><domain_name></domain>
</kerberos_sso>
```

Configuring Kerberos fallback

The Kerberos SSO Authentication Scheme provides the option to fall back to the default login mechanism to the web-application, on failure conditions. Set the <docbase_login_fallback> tag in the <kerberos_sso> tag in wdk/app.xml, to support the default login to the web-application, as follows:

```
<docbase_login_fallback>true</docbase_login_fallback>
```

The default value of the <docbase_login_fallback> tag is false.

Copy the <docbase_login_fallback> element into the <kerberos_sso> tag in custom/app.xml.

Sample Kerberos configuration in app.xml

The following code snippet is an example of the final configuration for Kerberos in app.xml.

```
Example 7-1. Code snippet in the wdk/app.xml file to enable Kerberos authentication

<authentication>
<!-- Kerberos SSO authentication scheme configuration -->
<kerberos_sso>
  <enabled>true</enabled>
  <browsers>
    <windows>
```

EMC Documentum Web Development Kit and Webtop Version 6.8.2 Deployment Guide
Configuring User Authentication

```xml
<ieversions><supported Internet Explorer browsers></ieversions>
<firefoxversions><supported Firefox browsers></firefoxversions>
</windows>

<!-- Enable login fall back to DocbaseLogin scheme -->
<docbase_loginFallback>false</docbase_loginFallback>
<!-- Mandatory configuration: Provide the kerberos realm / domain name. -->
<domain>WDKBLR.COM</domain>
</kerberos_sso>
</authentication>
```

Copy the `<authentication>` tag from the wdk/app.xml file into the custom/app.xml file.

Preparing the WDK application and browser to meet Kerberos Setup Requirements

This section discusses the setup requirements to enable Kerberos single sign-on authentication. Ensure that the client machine is already configured to use Kerberos authentication before you prepare the system for enabling Kerberos-based authentication.

Create user account for the WDK application in the active directory

You must register the WDK application as a Kerberos principal in the active directory to enable the WDK application to participate in Kerberos authentication. A Kerberos principal is a regular account on an Active Directory. The name of the principal can be something like the following: `name@YOUR.REALM`. The realm name follows the @ character in the principal. The principal represents the WDK application's service in the Kerberos realm.

To create a user in active directory:

   The Active Directory Users and Computers console is started.
2. Click a domain name and expand the contents.
3. Right-click Users and select New > User.
4. Type the user name in the Full Name field and in the Logon Name field.
5. Click Next.
6. Enter the password. Ensure that none of the password options are selected.
7. Click Next.
8. Click Finish.
9. Select the Users node in the left navigation bar of the Active Directory Users and Computers console.
10. Select and right-click the user that you created, and select Properties.
11. Select one or both of the following encryption algorithms under Account options, in the Account tab, based on the encryption algorithms you require:
   - Use DES encryption types for this account
   - This account supports Kerberos AES 128 bit encryption

12. To enable delegation for a WDK application user account, see To enable delegation for a WDK application’s user account; page 55

The Delegation tab appears when you select Properties in the context menu of a user account, in the Active Directory Users and Computers console, only after you register the WDK application’s SPN to the user.

Define a Service Principal Name for the WDK application and create the KeyTab file

A Service Principal Name (SPN) is a unique name that identifies an instance of a service and is associated with the login account under which the service instance runs. Windows 2008 account names are not multi-part as Kerberos principal names. As a result, administrators cannot directly create an account of the name HTTP/hostname.dns.com. Such a principal instance is created using service principal name mappings. In this case, an account is created with a meaningful name and hostname, and a service principal name mapping is added for HTTP/hostname.dns.com.

To use Kerberos after defining the SPN for the application server (on which the WDK application is deployed), the administrator must create a keytab (key table) file for the WDK application. The WDK application requires the keytab file to authenticate itself to the Key Distribution Center (KDC).

The administrator must use the ktpass.exe command-line tool to register the SPN as a security principal in the Windows Server Active Directory and to create a KeyTab file on the KDC. This ktpass.exe is bundled with Windows 2008 Resource Toolkit package and must be installed separately. Run ktpass.exe on the Active Directory Server machine and when the keytab file is generated move it to the wdkapp_installation/WEB-INF folder on the application server machine.

   ktpass /pass <password> -out <user-name>.keytab -princ <SPN> -crypto AES128-SHA1 +DumpSalt -ptype KR5_NT_PRINCIPAL /mapOp set /mapUser <user-name>

You can run the ktpass command with the following parameters:

   ktpass /pass <password> -out wdkapp.keytab -princ HTTP/wdkapp.dctmlabs.com@DCTMLABS.COM -crypto AES128-SHA1 +DumpSalt -ptype KR5_NT_PRINCIPAL /mapOp set /mapUser wdkuser

This command generates the wdkapp.keytab file on the Active Directory machine. Copy this file to the wdkapp_installation/WEB-INF folder on the application server machine.

To enable delegation for a WDK application’s user account:

1. Select the Users node in the left navigation bar of the Active Directory Users and Computers console.

2. Select and right-click the user created according to the procedure specified in the Create user account for the WDK application in the active directory, page 54 section, and select Properties.

3. Select Trust this user for delegation to any service (Kerberos only) in the Delegation tab.
Configuring the client browser to use the SPNEGO protocol

For Internet Explorer and Edge, in Internet Options, ensure that you specify the web address of the host name to enable SSO in the Add this Web site to the zone field and also ensure that the Enable Integrated Windows Authentication (requires restart) option is selected. After setting these options, restart the browser.

For Firefox, the network.negotiate-auth.trusted-uris and network.negotiate-auth.delegationuris preferences list the sites that are permitted to engage in SPNEGO Authentication with the browser. Type a comma-delimited list of trusted domains or URLs. For example, type http://wdkapp.dctmlabs.com. After setting these options, restart the browser.

For Chrome, ensure that you configure the following registry settings by specifying the server name to enable SSO. For example, navigate to the registry editor, HKEY_LOCAL_MACHINE > SOFTWARE > Policies > Google > Chrome. Add two strings with value names as AuthNegotiateDelegateWhitelist and AuthServerWhitelist. The value data for these strings should be appserver.domain.com or *.domain.com. After setting these options, restart the browser.

In Windows 7 and in Windows Server 2008 R2, the Data Encryption Standard (DES) encryption type (security settings) for Kerberos is disabled by default. If you log in to the WDK application from a client computer having Windows 7 or Windows Server 2008 R2 as the operating system, perform the following configuration.

To select the AES128, DES, and RC4 Kerberos encryption types:
1. In the Group Policy Management Console (GPMC), navigate to the Security Options node.
2. Double-click the Network security: Configure encryption types allowed for Kerberos option.
3. Select the DES_CBC_CRC,DES_CBC_MD5, RC4_HMAC_MD5, and AES128_HMAC_SHA1 encryption types.

Creating the JAAS configuration file

The KerberosSSOAuthenticationScheme class uses the Java JAAS and GSS-API to perform Kerberos authentication. The administrator must create the JAAS configuration file in the wdk_app_app_root_directory/WEB-INF folder; for example, wdk_app_root_directory/WEB-INF/krb5Login.conf.

Create the JAAS configuration file as follows:

```
<loginContext>
{
   <LoginModule> required
      debug=true
      principal="<SPN>"
      realm="<REALM>"
      refreshKrb5Config=true
      noTGT=true
      useKeyTab=true
      storeKey=true
      doNotPrompt=true
      useTicketCache=false
      isInitiator=false
      keyTab="<wdk_app_user_keytab_path>";
```
Configuring User Authentication

where:

| **<loginContext>** | Corresponds to the WDK application’s SPN. You replace separator characters with hyphen characters and omit the @REALM segment in the SPN. For example, the following LoginContext is derived from the corresponding SPN:
|                | • LoginContext:
|                |   HTTP-wdkapps-wdkblr-com
|                | • SPN:
|                |   http/wdkapps.wdkblr.com@WDKBLR.COM
|                | **Note:** Make sure that the SPN in the JAAS configuration matches the SPN defined in web.xml. |

| **<LoginModule>** | Specify the Kerberos login module to be used to perform user authentication:
|                | • For single-domain support only:
|                |   com.sun.security.auth.module.Krb5LoginModule
|                | • For both multi- and single-domain support:
|                |   com.dstc.security.kerberos.jaas.KerberosLoginModule
|                | **Note:** This module is the Quest KerberosLoginModule. |

| **<SPN>** | The WDK application’s SPN.
|           | For example, for single-domain support:
|           |   http/wdkapps.wdkblr.com@WDKBLR.COM
|           | For multi-domain support, instead of appending the domain name to the SPN, use the realm property to specify the domain name. |

| **<REALM>** | (Multi-domain support only) The realm name.
|             | For example: WDKBLR.COM |

| **<wdk_appu_user_keytab_path>** | The path to the WDK application user account’s *.keytab file in the WEB-INF folder of Tomcat. For example:<wdk_app_root>/WEB-INF/xxx.keytab |
Creating a configuration file for the application server to connect to the KDC server

To specify the KDC server to which the application server connects, create a configuration file in %WINDIR% (Windows), /etc (Linux), or /etc/krb5 (AIX). The names of the configuration files are krb5.ini (Windows) and krb5.conf (Linux and AIX). Refer to the following examples.

Example 7-2. Data Encryption Standard (DES) as a permitted encryption type

```
[libdefaults]
default_realm = WDKBLR.COM
forwardable = true
ticket_lifetime = 24h
clockshek = 72000
default_tkt_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-shal
default_tgs_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-shal
permitted_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-shal

[realms]
  WDKBLR.COM= { 
    kdc = WDKWIN5175.WDKBLR.COM
    admin_server = WDKWIN5175.WDKBLR.COM
  }
```

Example 7-3. Both DES and AES as permitted encryption types

To specify the Advanced Encryption Standard (AES) as a permitted encryption type along with DES:

```
[libdefaults]
default_realm = <Kerberos_domain_name>
forwardable = true
ticket_lifetime = 24h
clockshek = 72000
default_tkt_enctypes = aes128-cts rc4-hmac des3-cbc-shal des-cbc-md5 des-cbc-crc
default_tgs_enctypes = aes128-cts rc4-hmac des3-cbc-shal des-cbc-md5 des-cbc-crc
permitted_enctypes = aes128-cts rc4-hmac des3-cbc-shal des-cbc-md5 des-cbc-crc

[realms]
  <Kerberos_domain_name>= { 
    kdc = <KDC_server_address>
    admin_server = <KDC_server_address>
  }
```

Modify the Windows configuration file with the following details:

- Specify the Kerberos domain name as the default_realm.
- The realms section points to the KDC server.
Application server configuration

You must configure the application server on which the WDK application is deployed as described in the following sections:

- **Tomcat**, page 59
- **WebLogic**, page 59
- **JBoss**, page 59
- **WebSphere**, page 60

**Tomcat**

In `Tomcat_home_directory/bin/Catalina.bat` or `catalina.sh`, set the following JAVA options:

```bash
set JAVA_OPTS=% JAVA_OPTS % -Djava.security.krb5.conf=<location of krb5.ini>
-Djava.security.auth.login.config=<location of krb5Login.conf>
-Djavax.security.auth.useSubjectCredsOnly=false
```

**WebLogic**

In `WebLogic_home_directory/user_projects/domains/your_domain/bin/setDomainEnv.cmd` or `setDomainEnv.sh`, set the following JAVA options:

```bash
set JAVA_OPTIONS=%JAVA_OPTIONS% -Xms256m -Xmx1024m -Xdebug -Xnoagent
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
-Djava.security.krb5.conf=<location of krb5.ini>
-Djava.security.auth.login.config=<location of krb5Login.conf>
-Djavax.security.auth.useSubjectCredsOnly=false
```

**Note:** The default location of the `krb5.ini` file is `%WINDIR%` (Windows).

**JBoss**

**JBoss 5.x**

In `JBoss_home_directory/Server/server_name/login-config.xml`, add the following configuration:

```
<application-policy name="HTTP-hostName-realm_Name">
<authentication>
    <login-module code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
        <module-option name="debug">true</module-option>
        <module-option name="storeKey">true</module-option>
        <module-option name="useKeyTab">true</module-option>
        <module-option name="keyTab">path_to_keytab_file</module-option>
    </login-module>
</authentication>
</application-policy>
```
Example 7.4. Sample configuration for JBoss

```xml
<application-policy name="HTTP-wdkapps-wdkblr-com">
  <login-module code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
    <module-option name="principal">SPN</module-option>
  </login-module>
  <module-option name="debug">true</module-option>
  <module-option name="storeKey">true</module-option>
  <module-option name="keyTab">C:/jboss-eap-4.3/jboss-as/server/production/deploy/webtop.war/WEB-INF/kerberosas.keytab</module-option>
  <module-option name="principal">HTTP/kerberosas.wdkblr.com@WDKBLR.COM</module-option>
</login-module>
</application-policy>
```

JBoss 6.3.x

In `JBoss_home_directory\standalone\configuration\standalone.xml`, add the following configuration:

```xml
<security-domain name="HTTP-wdkapps-wdkblr-com">
  <login-module code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
    <module-option name="debug">true</module-option>
    <module-option name="storeKey">true</module-option>
    <module-option name="keyTab">C:/JBoss_home_directory/jboss-as/server/production/deploy/webtop.war/WEB-INF/kerberosas.keytab</module-option>
    <module-option name="principal">HTTP/kerberosas.wdkblr.com@WDKBLR.COM</module-option>
    <module-option name="useTicketCache">false</module-option>
    <module-option name="doNotPrompt">true</module-option>
    <module-option name="noTGT">true</module-option>
  </login-module>
</security-domain>
```

For application servers running in managed domains, add the configuration in `JBoss_home_directory\domain\configuration\domain.xml`. The application server documentation contains more details. The *EMC Documentum Foundation Classes Development Guide* contains detailed information on Kerberos secure SSO.

WebSphere

- In `WebSphere_home_directory\AppServer\profiles\AppSrv01\properties\wsjaas.conf`, add the following configuration:

```
HTTP-hostName realm_Name { com.ibm.security.auth.module.Krb5LoginModule
  required debug=true credsType="both" useKeytab="file:fullPathToKeytabfile"
```
principal="HTTP/hostname.realmName";

- Create a configuration file to specify the KDC server to which the application server should connect, in the `%WINDIR%` (Windows) or in `/etc/krb5` (AIX). The names of the configuration files are `krb5.ini` (Windows) and `krb5.conf` (AIX). To support Advanced Encryption Standard (AES) in the WebSphere Application Server, specify `aes128-cts-hmac-sha1-96` as a permitted encryption type.

Example 7-5. Both DES and AES as permitted encryption types

```plaintext
[libdefaults]
default_realm = WDKBLR.COM
forwardable = true
ticket_lifetime = 24h
clockskew = 72000

default_tkt_enctypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-shal
des-cbc-md5 des-cbc-crc
default_tgs_enctypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-shal
des-cbc-md5 des-cbc-crc
permitted_enctypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-shal
des-cbc-md5 des-cbc-crc

[realms]
  WDKBLR.COM= {
      kdc = WDKWIN5175.WDKBLR.COM
      admin_server = WDKWIN5175.WDKBLR.COM

  }
```

**Configuring Netegrity SiteMinder SSO authentication**

Netegrity Server is a high-performance, multi-threaded web server that applies fine-grained security policy to a protected network. EMC Documentum can integrate with Netegrity Server, its SSO solution, or any other SSO solution supported by Netegrity Server.

For more information about installing and configuring the Netegrity Server, see the related Netegrity Server documentation. For more information about configuring Documentum applications to enable Netegrity SSO authentication, see the Development Guide or Installation Guide of your application.

WDK applications support SSO using Netegrity SiteMinder. Netegrity SiteMinder users must have the same login names as the Content Server repository.
Prerequisites

- Set the precedence of authentication schemes in the `com.documentum.web.formext.session.AuthenticationSchemes.properties` file. For more information, see the EMC Documentum Web Development Kit Development Guide.
- Install and configure the Netegrity SSO environment, and create a HTTP or HTTPS junction that will link the Netegrity Server to Webtop.
- Deploy Webtop on the application server machine, and connect to a Content Server that has been configured for Netegrity SSO authentication. See Chapter 5, Deploying a WDK-based Application for more information to deploy Webtop on an application server. For more information about configuring Content Server for Netegrity SSO authentication, see the EMC Documentum Content Server Installation Guide, EMC Documentum Content Server Administration and Configuration Guide, and the EMC Documentum Administrator User Guide.

Enabling single sign-on (SSO)

1. Configure the Netegrity SiteMinder security server to authenticate repository users.
2. Configure the web application server to use an external HTTP Server supported by the Netegrity Server.
3. Configure the Content Server plugin for Netegrity SiteMinder SSO. The EMC Documentum Content Server Administration and Configuration Guide contains more information.
5. Locate the file AuthenticationScheme.properties in WEB-INF/classes/com/documentum/web/formext/session. Modify the properties file to make Netegrity SSO scheme (SSOAuthenticationScheme) as first in the list of authentications that are attempted during login. If the Docbase Login scheme is listed before the SSO scheme, the user is presented with a login screen instead of single sign-on.
6. Restart the application server.

Configuring app.xml for Netegrity Server single sign-on

1. Open the app.xml file in your applications/custom directory.
2. Copy from app.xml the `<authentication>` element and its entire contents, and paste into your custom app.xml.
3. Update the `<sso_config>` element under the existing `<authentication>` element as shown in the following example:

```
<authentication>
  <domain/>
  <docbase>secure_docbase</docbase>
  <service_class>
    com.documentum.web.formext.session.AuthenticationService
  </service_class>
```
Table 3. Authentication elements (<authentication>)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;docbase&gt;</td>
<td>Specifies default repository name. When SSO authentication is enabled but a repository name is not explicitly spelled out by the user nor defined in this element, the sso_login component is called. In this case the component prompts the user for the repository name.</td>
</tr>
<tr>
<td>&lt;domain&gt;</td>
<td>Specifies Windows network domain name.</td>
</tr>
<tr>
<td>&lt;service_class&gt;</td>
<td>Specifies fully qualified name of class that provides authentication service. This class can perform pre- or post-processing of authentication.</td>
</tr>
<tr>
<td>&lt;sso_config&gt;</td>
<td>Contains SSO authentication configuration elements.</td>
</tr>
<tr>
<td>&lt;sso_config&gt;.&lt;ecs_plug_in&gt;</td>
<td>Specifies name of the Content Server authentication plugin (not the authentication scheme name). Valid value: dm_netegrity</td>
</tr>
<tr>
<td>&lt;sso_config&gt;.&lt;ticket_cookie&gt;</td>
<td>Specifies name of vendor-specific cookie that holds the sign-on ticket. Valid value: SMSESSION</td>
</tr>
<tr>
<td>&lt;sso_config&gt;.&lt;user_header&gt;</td>
<td>Specifies name of vendor-specific header that holds the username. Valid value: SM-USER.</td>
</tr>
</tbody>
</table>
Configuring User Principal authentication

The EMC Documentum Web Development Kit Development Guide contains detailed information.
Chapter 8

Installing Application Connectors

This chapter describes how to install Application Connectors.

Overview

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

For information about software and hardware requirements, including supported Microsoft Office applications and versions, see the EMC Documentum Environment and System Requirements Guide or the product release notes document.

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

- GUI installation
  
  The administrator notifies the end user to install Application Connectors. The email contains the URL to the installer. The installer is part of the WDK application, in the path /webcomponent/install/appconnectors.

  The path is changed to /webcomponent/install/appconnectors/32bit (for 32-bit) and /webcomponent/install/appconnectors/64bit (for 64-bit).

- Command-line installation
  
  Microsoft Systems Management Server (SMS) is used to distribute Application Connectors to Microsoft Office users with an Microsoft Installer (MSI) based installer.

⚠️ **Caution:** Do not install Application Connectors using the MSI file extracted from the Documentum-AppConnectors-Client.exe file.

Application Connectors work with UCF content transfer only.

The Application Connectors installer disables Documentum Desktop Office integrations before installing Application Connectors. The Desktop Office integrations are disabled by removing relevant add-in files and registry entries.
The installer executable is the Documentum-AppConnectors-Client.exe file. When the installer is run, it verifies on each client host that the following requirements are met:

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

During installation, Application Connectors also checks for a supported 32-bit JRE version running on the client. If no suitable JRE is found, Application Connectors unzips and installs its own private JRE into the <Program Files>/Documentum/AppConnector/Jre folder and uses this for all content transfer operations.

In case you change or uninstall the Java version after installing Application Connectors, you can run the JavaPathUpdater.vbs file in <Program Files>/Documentum/AppConnector. This will ensure that the latest Java changes are used and updates run.bat with the most suitable JRE version.

The Application Connectors installer does not perform an in-place upgrade to an existing installation on the client. You must uninstall the previous version and delete the entire installation directory.

Note: Before you uninstall, ensure that you have terminated the UCF process (javaw.exe, if running).

**GUI installation of Application Connectors**

Ensure that the Webtop-based application is running and available when you run the Application Connectors installer.

**To download and install Application Connectors on the client host:**

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

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

   ```
   http://hostname:port/webtop/webcomponent/install/
   appconnector/<32bit> or <64bit>/Documentum-AppConnectors-Client.exe
   ```

5. Click **Install** and follow the wizard.
6. In the Customer Information dialog, choose **Current User** or **All Users**.
7. For **Enter URL**, type the URL to the Webtop-based installation that you will use within the Office applications, for example:

   ```
   http://plelegion:8080/webtop
   ```
After installation has completed, the Documentum menu is available within the authoring application for which an Application Connector was installed.

- Microsoft Word/Excel/PowerPoint: **File > Add-Ins > Documentum**
- Microsoft Outlook: **Add-Ins > Documentum**

**Note:**

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

## Command-line installation of Application Connectors

The installer is located within the Webtop-based application in the folder `/webcomponent/install/appconnector`.

For example: The path is `/webcomponent/install/appconnector/32bit` (for 32-bit) and `/webcomponent/install/appconnector/64bit` (for 64-bit)

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

**Running the installer from the command line** — Here is the syntax to run the installer in command-line mode:

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

**Running the installer in silent mode** — The following syntax launches the installer silently from the command line:

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

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

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

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

```
Documentum-AppConnectors-Client.exe /s /v"/qn WEBTOPURL=http://server/webtop DELETE_NORMAL_DOT_DOT=TRUE"
```
Chapter 9

Enabling the Webtop Express DocApp

Content Server 6 SP1 or higher installs the Webtop Express DocApp. This DocApp creates lightweight functionality for an Express user by means of presets. To make this functionality available, add users to the Express User (express_user) role. This role is installed by the Webtop Express DocApp.

Table 4, page 69 describes the functionality that is available to Webtop Express users.

Table 4. Express user capabilities

<table>
<thead>
<tr>
<th>Preset</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formats</td>
<td>None</td>
</tr>
<tr>
<td>Types</td>
<td>dm_document</td>
</tr>
<tr>
<td>Templates</td>
<td>Displays templates that correspond to formats</td>
</tr>
<tr>
<td>Actions</td>
<td>Document: Content transfer, subscriptions, email, quickflow, Properties, clipboard actions, create, delete</td>
</tr>
<tr>
<td></td>
<td>Excluded: Relationships, export to CSV, favorites, notifications, lifecycle and virtual document actions, tools (most); new workflow template, room, form, cabinet</td>
</tr>
<tr>
<td>Locations</td>
<td>Home Cabinet</td>
</tr>
</tbody>
</table>

Presets administrators who belong to the dmc_wdk_presets_coordinator role can change the enabled or excluded features and allowable values by editing the Webtop Express preset in the Presets Editor UI.
This chapter contains information on troubleshooting a WDK application deployment. Not all items may apply to your WDK-based product or environment.

Wrong JRE used for application server

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

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

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

No global registry or connection broker

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

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

To fix this error, either provide the correct BOF registry connection information in dfc.properties, or do not provide any connection information at all. Refer to the *EMC Documentum Content Server Installation Guide* for information on enabling a repository as a global registry.
No connection to repository

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

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

A WDK-based application must have information about the available connection broker in order to establish a connection to repositories. Refer to To configure connections in dfc.properties before deployment, page 34 for information on enabling the connection in dfc.properties.

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

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

DM_VEL_INSTANTIATION_ERROR

This error can be caused by several setup problems:
- Not using a version 6 global registry repository
- Installing DAB 5.3 on the same machine as the application server

Login page incorrectly displayed

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

If the login page displays several controls with the same label, you have not turned off tag pooling in the application server. Refer to Tag pooling problem, page 73 for troubleshooting information on this problem.

Slow performance

Many performance enhancements are documented in EMC Documentum Web Development Kit Development Guide.

Set dfc.diagnostics.resources.enable to false in dfc.properties unless you are using the DFC diagnostics. This setting uses a significant amount of memory.
Out of memory errors in console or log

Check to make sure that you have allocated sufficient RAM for the application server VM. For more information, refer to Setting the Java memory allocation, page 25.

The following error is common when the MaxPermSize is set too low:
java.lang.OutOfMemoryError: PermGen space

Slow display first time

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

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

DFC using the wrong directories on the application server

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

Application startup errors

If you installed a WDK-based application of version 5.x, it has modified your application server startup file. Run the WDK-based application uninstaller to remove these modifications. Modifications to the start script are no longer required by WDK 6.x or higher. Each WDK-based application contains the libraries required for version 6.x within the WEB-INF directory.

You must also verify that your application server host does not set environment variables for the JRE location which will cause the application to use the wrong JRE.

Tag pooling problem

If you have not properly disabled tag pooling in the application server, you will see several instances of the same control on the login page. For instructions on disabling pooling in Tomcat, refer to Preparing Tomcat, page 29.

Caution: After you disable tag pooling, you must clear the cached JSP class files which still may contain pooled tags. Refer to your application server documentation to find the location of the generated class files. For example, Tomcat displays the following error message:
com.documentum.web.form.control.TagPoolingEnabledException: JSP tag pooling is not supported.

UCF client problems

If the error message "Compatible Java Run time environment is not installed" is displayed on a non-Windows client, make sure that you have installed a supported version of the Oracle JRE on the client; this version will be used by UCF and will not interfere with the browser VM. The client browser VM must be one that is certified in the EMC Documentum Environment and System Requirements Guide or the product release notes document. It will be used for non-UCF applets.

If a UCF error is reported on the client, the following troubleshooting steps may help:

- For UCF timeouts, check whether anti-virus software on the application server is monitoring port 8080 or the application server port that is in use. You may need to turn off monitoring of the application server port.
- For very slow UCF downloads, check to make sure virus scanning within zip files is not turned on.
- Ensure that the user has a supported JRE version on the machine in order to initiate UCF installation. You can point the client browser to a Java tester utility such as Javatester utility to verify the presence and version of a JRE.
- See if the process from the launch command is running: Open the browser Java console look for "invoked runtime: ... connected, uid: ... A UID indicates successful connection to the UCF server.
- Are there any errors on the UCF server side? Check the application server console.
- Restart the browser and retry the content transfer operation.
- Kill the UCF launch process and retry the content transfer operation.
- If UCF operations still do not launch, delete the client UCF folder located in USER_HOME/username/Documentum/ucf.
- Search the client system for files that start with ucfinit.jar- and delete them.
- Delete the JRE cache from the JRE Control Panel > Temporary Internet Files.
- Delete the proxy server cache.

Citrix client problems

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

To configure the web application for roaming profiles

1. Delete the documentum directory that was installed in the user’s home directory, for example, C:\Documents and Settings\<user name>\Documentum.
2. Edit ucfinstaller.config.xml in /wdk/contentXfer in the WDK application. Change every environment variable in this file that uses the Java home directory $java[user.home] to use the roaming profile environment variable:

   <defaults>
   <ucfHome value="$env(USERPROFILE)/Documentum/ucf"/>
   <ucfInstallsHome="$env(USERPROFILE)/Documentum/ucf"/>
   <configuration name="com.documentum.ucf">
      <option name="user.dir">
      <value>$env(USERPROFILE)/Documentum</value>
   </option>
   
3. Save and restart the application server.

**Connection issues between an Federated Search server and IPv6 clients**

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

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

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

**Presets not working**

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

**Blank page error on deploying DA**

Deploying DA on a WebSphere 6.1 environment throws a blank page due to classloading constraint violation. To resolve this, add a new property to dfc.properties as below:

```
With PARENT_LAST, dfc.bof.classloader.enable_extension_loader_first = false
With PARENT_FIRST, dfc.bof.classloader.enable_extension_loader_first = true
```
Documentum Application Connectors refers to Microsoft Virtual Machine installed on the client while calling UCF methods

If a Microsoft Virtual Machine is installed on the client machine, then Documentum Application Connectors uses it while calling UCF methods, resulting in exceptions.

Set the following environment variable to ensure Oracle JRE is used:

```
JAVA_PLUGIN_WEBCONTROL_ENABLE = TRUE
```
Chapter 11

Deploying a Custom Application

Using the comment stripper utility

Your JSP pages will load faster if you strip out white space and comments. A comment stripper tool, CommentStripper, is provided in /WEB-INF/classes/com/documentum/web/tools. Table 5, page 77 describes the parameters to use in starting this tool from the console.

Table 5. Comment stripper utility parameters

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

The commented files, useful for development, are provided in a JAR file in the base directory: unstripped.jar.
Appendix A

Predeployment Checklist

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

Table 6. Predeployment tasks

<table>
<thead>
<tr>
<th>Requirement</th>
<th>For More Information</th>
<th>Completed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review the release notes for the release you are installing or to which you are upgrading.</td>
<td>The release notes are available on the EMC Documentum download site.</td>
<td></td>
</tr>
<tr>
<td>Validate your hardware configuration.</td>
<td>EMC Documentum Environment and System Requirements Guide or product release notes</td>
<td></td>
</tr>
<tr>
<td>Validate your application server and clients operating systems.</td>
<td>EMC Documentum Environment and System Requirements Guide or product release notes</td>
<td></td>
</tr>
<tr>
<td>Create any required operating system accounts.</td>
<td>Network administrators</td>
<td></td>
</tr>
<tr>
<td>Verify that the application server instance owner has write permissions on the temporary content transfer directories.</td>
<td>Network administrators. The requirement is described in Content transfer directory permissions, page 15.</td>
<td></td>
</tr>
<tr>
<td>Determine the repositories to which end users of the application will connect.</td>
<td>Network administrators</td>
<td></td>
</tr>
<tr>
<td>Determine the connection brokers to which the repositories project.</td>
<td>Network administrators</td>
<td></td>
</tr>
<tr>
<td>Determine which repository on the network is the global registry repository, and obtain the global registry user’s user name and password.</td>
<td>Network administrators</td>
<td></td>
</tr>
</tbody>
</table>

EMC Documentum Web Development Kit and Webtop Version 6.8.2 Deployment Guide 79
<table>
<thead>
<tr>
<th>Requirement</th>
<th>For More Information</th>
<th>Completed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine which repositories will be used to store presets and user preferences.</td>
<td>Network administrators</td>
<td></td>
</tr>
<tr>
<td>Determine whether language packs will be required.</td>
<td><em>EMC Documentum Web Development Kit Applications Language Pack Installation and Release Notes</em></td>
<td></td>
</tr>
<tr>
<td>Prepare the application server host and application server software according to the vendor’s requirements.</td>
<td>Specific requirements are described in <em>Chapter 4, Preparing the Application Server Host.</em></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

Directories and Files to Back Up Before Migration

For complete information about migration, see the EMC Documentum System Upgrade and Migration Guide.

Table 7. Directories and files to back up

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