# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Chapter 1</td>
<td>Quick Start</td>
<td>11</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Planning for Deployment</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Required and optional supporting software</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Typical configuration</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Preparing the Content Server</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Application server host requirements</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Directory name restriction</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Content transfer directory permissions</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>DNS resolution</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Deploying multiple applications</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Deploying language packs</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Customizing an application</td>
<td>16</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Planning for Mixed Environments (5.3.x and 6.x)</td>
<td>17</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Preparing the Client Hosts</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Ensuring a certified JVM on browser clients</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Enabling HTTP content transfer in Internet Explorer 7</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Enabling UCF content transfer in Internet Explorer 7 on Windows Vista</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Enabling content transfer in Firefox</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Firefox version 3.x</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Supporting Outlook mail message archives</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Using Citrix Presentation Server Client</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Turning off the pop-up blocker in Internet Explorer</td>
<td>25</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Preparing the Application Server Host</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Setting the Java memory allocation</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Turning off failover</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Preparing environment variables for non-default DFC locations</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Preparing JBoss and Apache Tomcat</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Preparing WebLogic server</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Supporting large content transfer operations in a managed server environment</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Preparing IBM WebSphere</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Supporting failover in a cluster</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Applying policies for WebSphere security</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Preparing Oracle Application Server</td>
<td>32</td>
</tr>
</tbody>
</table>
# Table of Contents

Preparing Sun Java System Application Server ............................................. 33  
Turning off tag pooling ....................................................................................... 33  
Modifying the Sun policy file ............................................................................ 33  
Turning off failover ............................................................................................. 34  
Preparing to use an external web server ............................................................ 34  

**Chapter 6**  
**Migrating a WDK-Based Application** ............................................................. 37  

**Chapter 7**  
**Deploying a WDK-Based Application** .......................................................... 39  
Preparing the WAR file for deployment .............................................................. 39  
Enabling DFC connections to repositories ......................................................... 40  
Enabling DFC memory optimization ................................................................. 41  
Configuring UCF ................................................................................................. 42  
How to configure JRE using UCF ........................................................................ 42  
Enabling presets and preferences repositories ................................................. 43  
Enabling retention of folder structure and objects on export ............................. 43  
Enabling modal pop-up windows ....................................................................... 43  
Enable the EMCMF format in WDK-based applications ................................. 44  
Enabling external searches ................................................................................ 45  
Configuring the connection to the search server ............................................. 45  
Configuring the connection to the backup search server ................................... 46  
Deploying multiple applications ....................................................................... 46  

**Chapter 8**  
**Completing the Deployment** ....................................................................... 47  
Configuring IBM WebSphere after deployment ............................................... 47  
changing the classloader and compiler settings .............................................. 47  
Setting com.ibm.ws.webcontainer.invokefilterscompatibility to true .......... 48  
Configuring Oracle Application Server ......................................................... 48  
Deploying default virtual link support ............................................................. 48  
Accessing the application .................................................................................. 49  
Testing WDK samples ....................................................................................... 49  

**Chapter 9**  
**Configuring Single Sign-On for Security Servers** ....................................... 51  

**Chapter 10**  
**Installing Application Connectors** ............................................................... 55  
Overview ........................................................................................................... 55  
Enabling installation on Windows 2003 .......................................................... 56  
GUI installation of Application Connectors ...................................................... 56  
Command-line installation of Application Connectors ................................... 58  
Location of installed files on the client host .................................................... 59  

**Chapter 11**  
**Installing Documentum Webtop Federated Search Service (FS2)** .......... 61  
Installing the Webtop Federated Search Service (FS2) option .......................... 62  
Viewing installation log files ............................................................................ 63  

**Chapter 12**  
**Enabling the Webtop Express DocApp** ....................................................... 65  

**Chapter 13**  
**Troubleshooting Deployment** ................................................................... 67  
Wrong JRE used for application server ............................................................ 67  

EMC Documentum Web Development Kit and Webtop 6.5 SP1 Deployment Guide
| Chapter 14 | Deploying a Custom Application | 73 |
| Appendix A | Predeployment Checklist | 75 |
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.</td>
<td>Basic WDK host configuration</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2.</td>
<td>Documentum menu in authoring application</td>
<td>58</td>
</tr>
</tbody>
</table>
List of Tables

Table 1. Directories and files to back up ................................................................. 37
Table 2. Preferences configuration elements ............................................................... 43
Table 3. Modal window elements in app.xml (<modalpopup>) ........................................ 44
Table 4. Authentication elements (<authentication>) .................................................... 52
Table 5. Location of files installed by Application Connectors on the client host ............. 59
Table 6. Express user capabilities ................................................................................ 65
Table 7. Comment stripper utility parameters .............................................................. 73
Table 8. Predeployment tasks ..................................................................................... 75
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 based on industry standards 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. For additional information on developing or customizing applications with WDK, refer to the Web Development Kit Development Guide.

This guide may include instructions for application servers, or combinations of operating systems and application servers that are not supported for your product. For the list of supported platforms, refer to the release notes for the product that you are deploying.

**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. These client deployment guides contain the same general information that is in this guide as well as information specific to the client product.

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**

The following changes have been made to this document.

<table>
<thead>
<tr>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2008</td>
<td>Initial publication for version 6.5 SP1.</td>
</tr>
</tbody>
</table>
Related documentation

You can download this deployment guide and the release notes for WDK and Webtop on the product download site. Each Webtop-based application contains an online user guide in HTML format and a PDF version of that guide in the help subdirectory of the application WAR file.

The following developer documentation is available on the product download site for configuring and customizing WDK and Webtop applications. Additionally, some WDK-based products have their own development guides with information specific to the product.

- **Web Development Kit Development Guide**
- **Web Development Kit and Webtop Reference Guide**
- **Web Development Kit Tutorial**
- **System Upgrade and Migration Guide**
- **WDK_Samples_and_TestBed_version.zip**
  where *version* is the product version number.
- **JavaDoc API reference documentation**
This chapter describes the steps you need to perform to deploy your application. The steps are described in more detail in the chapters of this guide. Your product or environment may require additional steps, which you can find in the product-specific chapter or chapters of this guide or in the index.

To 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 4, Preparing the Client Hosts.)
   Install a supported browser virtual machine and perform specific browser preparations for IE 7 and Firefox. If needed, you will install the mail message converter and prepare Citrix clients.

3. Prepare the application server. (Refer to Chapter 5, 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 7, 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, optional presets and preferences repositories, and optional Federated Search server.

5. Complete the deployment. (Refer to Chapter 8, 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. Check your release notes for information on the application servers, browsers and other software in the environment that are certified for your product.

This chapter discusses the following topics:

- Required and optional supporting software, page 13
- Typical configuration, page 14
- Preparing the Content Server, page 15
- Application server host requirements, page 15
- Deploying multiple applications, page 16
- Deploying language packs, page 16
- Customizing an application, page 16

### Required and optional supporting software

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 41 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. The Webtop DARs are provided in Content Server version 6.5.
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.

Figure 1. Basic WDK host configuration

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. Refer to the release notes for each WDK-based application to learn which managed server configurations are supported.
Preparing the Content Server

The following topics describe Content Server requirements.

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 6.x must be installed in your environment in order to run a WDK-based application. A global registry is a Content Server that has been designated as a global registry. For information on designating your application’s global registry before deployment, refer to Enabling DFC connections to repositories, page 40.

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 Web Development Kit Development Guide.

Some application servers require policies that grant permissions to write to these directories. Refer to deployment information for your application server to see 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 6.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 6.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 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.
Chapter 3

Planning for Mixed Environments (5.3.x and 6.x)

All WDK-based applications require DARs or DocApps that must be installed in the repository. The Webtop 6.5 DARs or 6.0 DocApps are provided in the Content Server of the corresponding version. If your application supports connections to a Content Server version 5.3.x, you must have a Content Server 6.x global registry.

The following features in the Webtop DARs or DocApps will not be available with a 5.3.x Content Server:

- Accelerated Caching Services (ACS) and Branch Office Caching Services (BOCS) write operations
- Create relationships
- Presets
- Lifecycle enhancements in the properties, checkin, import and new document user interface.
- LWSOs

If the Content Server version is 5.3, the read notification feature will use the 5.3 email notification script that is configured in the Content Server. It will not provide metadata or links.

**Preparing a 5.3.x Content Server repository** — The WDK-based application WAR file contains scripts to upgrade the repository for subscriptions. Run the DQL script subscriptionInstall.dql that is located under the root web application directory, in webcomponent/install. Taxonomy Manager support scripts are located in the directory webcomponent/install/admin/tm.

**Supporting WDK 5.3.x and 6.x applications on the application server** — A 5.3.x application cannot share the same instance as a version 6.x application. One or both of the applications will not work properly.
The following behavior of EMCMF would be applicable for Webtop 6.5 SP1 and Content Server 5.3.x or 6.x combination:

- EMCMF support: This is available provided you have configured EMCMF support on the Content Server 5.3.x or 6.x. For this, you have to install the following DocApps:
  - MessagingApp DocApp
  - DCO DocApp
  - Collaboration DocApp
Chapter 4

Preparing the Client Hosts

This chapter contains instructions that are shared by all WDK-based products. Check your release notes for information on the browsers that are certified for your product.

This chapter contains information on the following predeployment tasks:
- Ensuring a certified JVM on browser clients, page 19
- Enabling HTTP content transfer in Internet Explorer 7, page 20
- Enabling UCF content transfer in Internet Explorer 7 on Windows Vista, page 20
- Enabling content transfer in Firefox, page 21
- Supporting Outlook mail message archives, page 24
- Using Citrix Presentation Server Client, page 25

Ensuring a certified JVM on browser clients

Browser client hosts require a certified version of the Sun Java virtual machine (JVM or VM) to initiate content transfer in a WDK application. New machines may not have a JVM installed in the browser. Check the release notes for your product version for the JVMs that are supported.

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

On Windows clients, if the JVM required for UCF is not present on the client machine, UCF uploads to a Windows client a private JVM. This VM does not replace the JVM that is used by the browser. For non-Windows browser hosts with a JVM of 1.4.x, you must pre-install version 1.5.0_06.x of the Sun JRE that will then be used by UCF.

Since the UCF VM file (Sun 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.
Preparing the Client Hosts

Enabling HTTP content transfer in Internet Explorer 7

Internet Explorer (IE) version 7 has a default security setting that prevents the display of the file download dialog. You must add the WDK-based application URL to the list of trusted sites in the browser in order to perform checkout, view, or edit in HTTP mode.

Nothing happens when user exports as CSV if the browser security settings are disabled for ‘prompt for file downloads’ and ‘file download’. These are disabled by default in IE7. The user must enable them.

To enable HTTP file download in IE 7:
1. In the IE 7 browser menu, choose Tools > Internet Options and click the Security tab.
2. Choose Trusted sites and then click Custom level.
4. Click OK twice to save settings.
5. Close all browser windows and restart the browser.

Enabling UCF content transfer in Internet Explorer 7 on Windows Vista

Internet Explorer 7 on Windows Vista OS does not display a file download dialog to permit UCF content transfer unless it is enabled by adding the application server host to the trusted sites list and doing one of the following:

• Turn off User Account Control (UAC) security for each client.
• Configure the application to use file registry mode.

To add the application server host to the list:
1. In IE7, go to Tools > Internet Options Security tab.
2. Select Trusted sites. Click Custom level in the section Security level for this zone.
3. Scroll to Downloads and check Automatic prompting for file downloads.
4. Click OK to accept changes, and close the browser.

To turn off UAC on each client:
1. Log in as a user who has administrator privileges on the Windows Vista machine.
2. Open the Control panel and choose Administrative Tools.
3. In the left pane, choose User Accounts.
4. Choose Turn User Account Control on or off.
5. Uncheck Use User Account Control (UAC) to help protect your computer.
Preparing the Client Hosts

6. Click OK and restart the system.

**To configure UCF to use file registry mode:**

1. Ensure the clients have checked in all checked out files.

2. Open the file ucf.installer.config.xml located in WDK-based applications directory /wdk/contentXfer.

3. Locate the element `<platform os="windows" arch="x86"/>`, which configures Windows clients.

4. Locate the child element `<defaults><configuration name="com.documentum.ucf">.<option name="registry.mode">.

5. Change the value element to the following:
   `<value>file</value>`

6. Save and restart the application.

**Enabling content transfer in Firefox**

Firefox 2 requires a setting to enable content transfer.

**To enable file download in Firefox:**

1. Open the Options menu in Firefox.

2. In the Main dialog Downloads section, enable Show the Downloads window when downloading a file and Close it when all downloads are finished.

3. Check Always ask me where to save files.

4. On the Tabs dialog, check New pages should be opened in: a new window.

5. On the Content dialog, check Load images automatically, Enable JavaScript, and Enable Java.

6. Install the Firefox add-on FireBug, which is available from mozilla.org.

**Firefox version 3.x**

In Firefox version 3.x, the file browse tag `<input type="file"…./>` returns the file name only as its default behavior. Since WDK based applications require the absolute file path, configuration is required. See this link for more information: https://developer.mozilla.org/en/Updating_web_applications_for_Firefox_3#File_upload_fields

**To configure a Firefox version 3.x browser:**

1. Add a new preference called `signed.applets.codebase_principal_support` and set its value to true. This configuration can be performed either by each end user or by administrators who can push the configuration to all end users. Both procedures are provided below.
a. For end user configuration:
   • Type `about:config` in the browser location bar. Press `Enter` and accept the warning message as shown below:

   ![Warning Message]

   • Search for `signed.applets.codebase_principal_support`; if the preference is found, ensure that its value is set to `true`. If the preference is not available, add the preference as shown below and set its value to `true`. Close the browser and then relaunch it, verifying that the preference is set to `true`.

   ![Add Preference]

   ![Preference Value]

   ![New Boolean Value]

22 EMC Documentum Web Development Kit and Webtop 6.5 SP1 Deployment Guide
Preparing the Client Hosts

b. For administrators wishing to push the configuration to all users on Firefox:
   - Create a file named user.js and add the following details to the file:
     ```javascript
     user_pref("signed.applets.codebase_principal_support",true);
     ```
   - Push the user.js file into the user profile location of Mozilla Firefox. For more information, consult the update available at: https://developer.mozilla.org/En/A_Brief_Guide_to_Mozilla_Preferences

For example, the profile directory may appear as C:\Documents and Settings\<username>\Application Data\Mozilla\Firefox\Profiles\k6uvinlv.default.

The user profile directory location for Mozilla Firefox differs depending on the operating system. For more information, see http://kb.mozillazine.org/Profile_folder.

2. The end user must allow this change when notified by the Internet Security alert:

![Internet Security Alert](image)

Supporting Outlook mail message archives

WDK supports viewing and exporting dm_message_archive objects. If your WDK-based product displays Outlook mail messages, read these instructions.

dm_message_archive objects can be viewed as HTML or as .msg files in Outlook. To view or export dm_message_archive objects as .msg files, the client must have an email converter installed. This converter can be automatically installed as part of the UCF download. However, the converter can take a long time to download and install. You can avoid this delay by deploying the ES1_MRE.exe installer using standard mechanisms, such as Microsoft Systems Management Server (SMS).

**To enable automatic download of the email converter:**

1. Uncomment the ES1_MRE.exe section in the app\wdk\contentXfer\ucfinstaller.config.xml file on the application server.

2. Specify `<messageArchive-support>` in wdk/app.xml.

You can run an email message migration utility that converts the .msg objects to a .emcmf objects. The emcmf object type adds email-specific attributes and attachment handling. This utility is available on the Powerlink site.

**Note:** The Webtop 6.5 SP1 email functionality will work with 5.3 or 6.0 Docbase only if you have DCO, Collaboration Services and Messaging App DocApps installed on this Docbase as these DocApps are available with DCO license. If you do not have DCO license, the Webtop 6.5 SP1 email functionality would not be supported on the Docbase.
Using Citrix Presentation Server Client

Citrix Presentation Server Client can be used as a web browser. Check the release notes for your WDK-based product to determine whether Citrix clients are supported for your product.

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.

Turning off the pop-up blocker in Internet Explorer

Windows XP SP2 and SP3 installs a pop-up blocker in Internet Explorer that is enabled by default. HTTP content transfer operations in WDK applications are prevented by the pop-up blocker. You must turn off the pop-up blocker for HTTP transfer.
Chapter 5

Preparing the Application Server Host

This chapter contains instructions that are shared by all WDK-based products. Check your release notes for information on the application servers that are certified for your product.

This chapter describes the tasks you must complete to prepare the application server host before deploying your WDK-based application.

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.

Tip: EMC recommends but does not require that you uninstall DFC 5.x and any application that uses DFC 5.x and then reboot before deploying an application based on WDK 6.x. For uninstall procedures, refer to the 5.x product documentation.

This chapter contains the following sections. Your selected application server and optional external web server must be certified for your product.

- Setting the Java memory allocation, page 28
  Required information for all application servers
- Turning off failover, page 28
- Preparing environment variables for non-default DFC locations, page 28
  Information for enterprise environments that do not use the default (recommended) DFC environment locations.
- Preparing JBoss and Apache Tomcat, page 29
- Preparing WebLogic server, page 29
- Preparing IBM WebSphere, page 30
- Preparing Oracle Application Server, page 32
- Preparing Sun Java System Application Server, page 33
- Preparing to use an external web server, page 34

EMC does not provide support for installing or running application servers. Refer to the documentation for each application server for instructions on installing, stopping, starting, and running the application server. Contact the application server vendor for technical support.
Setting the Java memory allocation

The minimum recommended Sun Java memory allocation values for application servers on a small system are the following:

- Xms1024m  -Xmx1024m

Application servers can slow down, throw exceptions, or crash with an application that has a large number of Java Server Pages. Set the MaxPermSize parameter to 128 or higher to avoid this problem.

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.

To achieve better performance, add these parameters to the application server startup command line:

-server
-XX:+UseParallelOldGC

-server must be the first parameter on the command line.

Performance will improve because the Java client VM is not suitable for long running server jobs and the default Java garbage collector cannot clean up the heap quickly enough—especially when the application server machine runs on multiple CPUs.

For more information on these settings, refer to Java documentation at the Sun web site (http://java.sun.com). More information on application server performance tuning and benchmarking for Documentum products is available from your EMC Documentum SE or EMC Documentum Consulting.

Turning off failover

If your application server and environment combination does not support failover, you can turn off failover in app.xml. Refer to your product release notes to determine whether failover is supported for your environment.

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

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

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

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
Preparing the Application Server Host

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 %BEA_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 and Apache Tomcat

Please refer to your product release notes to determine whether JBoss or Apache Tomcat is a supported application server for your product.

You must disable tag reuse in JBoss and Apache Tomcat in the web.xml file of the Tomcat/conf directory. The location of web.xml file in JBoss application is //server/default/deploy/jboss-web.deployer/conf/web.xml. Find the JSP servlet entry in web.xml. Add the enablePooling initialization parameter and set it to false:

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

Preparing WebLogic server

Please refer to your product release notes to determine whether BEA WebLogic is a supported application server for your product.

The following topic describes preparations that may be necessary before you deploy a WDK-based application.
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. For additional details, refer to BEA's documentation on Web Serve Plug-ins parameters.

Preparing IBM WebSphere

Please refer to your product release notes to determine whether IBM WebSphere is a supported application server for your product.

The following topics describe how to prepare the application server to support failover in a cluster, to apply policies for Java 2 security, and to support non-default content transfer locations. Refer to your product release notes to determine whether failover is supported for your application.

Supporting failover in a cluster

Failover in a clustered environment requires a setting. Set the NoAffinitySwitchBack custom property to true in the WAS cluster. For more information on this setting, refer to the WebSphere documentation.

Applying policies for WebSphere security

If WebSphere global security is enabled for the application server, by default it enables Java 2 security. Java 2 security requires security policies. You must apply the policies in the Documentum files app.policy, library.policy and was.policy. These files are provided by EMC Documentum on the download site in the compressed archive PolicyFiles.zip. These files contain the minimum set of policies that are required for the application to run without error. Add these policies to your existing files.

You must set up the environment variables that are referenced in these policies, and the application server instance owner must have write permission on these directories. Define the following environment variables:

• dfc.data.dir
  By default, the dfc.data.dir directory is the Documentum subdirectory of the directory that contains the application server executable.

• webtop.content.xfer
  Specifies the temporary content transfer directory on the application server. Must match the value in app.xml of the element <contentxfer><server><contentlocationwindows> or <contentlocationunix>.
The policy files in PolicyFiles.zip contain the minimum required policies for the dfc.data.dir directory. To add additional policies for non-default content transfer locations, add the following lines to library.policy. For each policy that you add, set up an environment variable that specifies the non-default location.

**Tip:** Select only the policies that are needed for your application.

**Policy for local user directory (non-default location)** — This policy is required if the user directory for the application server host machine is a non-default location. The default location is the same as the location specified by the dcf.properties key dcf.data.dir.

```java
permission java.io.FilePermission "${dfc.user}/**/", "read, write, delete";
permission java.io.FilePermission "${dfc.user}/*", "read, write, delete";
```

**Policy for checkout and export directories (non-default location)** — These environment variables must specify the same location as the value of the dcf.properties keys dcf.checkout.dir and dfc.export.dir. The default locations for these directories are checkout and export subdirectories of dfc.data.dir.

```java
permission java.io.FilePermission "${dfc.checkout}/**/", "read, write, delete";
permission java.io.FilePermission "${dfc.checkout}/*", "read, write, delete";
permission java.io.FilePermission "${dfc.export}/**/", "read, write, delete";
permission java.io.FilePermission "${dfc.export}/*", "read, write, delete";
```

**Policy for DFC registry file (non-default location)** — The value of the dcf.registry environment variable must match the location specified in dcf.properties for the key dcf.registry.file.

```java
permission java.io.FilePermission "${dfc.registry}/**/", "read, write, delete";
permission java.io.FilePermission "${dfc.registry}/*", "read, write, delete";
```

**Policy for Webtop temporary content transfer directory (non-default location)** —

```java
permission java.io.FilePermission "${webtop.content.xfer}/**/", "read, write, delete";
permission java.io.FilePermission "${webtop.content.xfer}/*", "read, write, delete";
```

**Policy for non-Webtop WDK-based temporary content transfer (non-default location)** — You can use this policy for TaskSpace or other application that is not based on Webtop:

```java
permission java.io.FilePermission "${wdk.content.xfer}/**/", "read, write, delete";
permission java.io.FilePermission "${wdk.content.xfer}/*", "read, write, delete";
```

**Policy for documentum applications directory (non-default location)** — The default location is dfc.data.dir.

```java
permission java.io.FilePermission "${documentum}/**/", "read, write, delete";
permission java.io.FilePermission "${documentum}/*", "read, write, delete";
```
Preparing the Application Server Host

Policy for DFC class cache directory (non-default location) — The default location is dfc.data.dir/cache.

permission java.io.FilePermission "${dfc.cache.dir}/**", "read, write, delete";
permission java.io.FilePermission "${dfc.cache.dir}", "read, write, delete";

Policy for Web Publisher —

permission java.io.FilePermission "${wp.content.xfer}/**", "read, write, delete";
permission java.io.FilePermission "${wp.content.xfer}", "read, write, delete";

Policy for Documentum Administrator —

permission java.io.FilePermission "${da.content.xfer}/**", "read, write, delete";
permission java.io.FilePermission "${da.content.xfer}", "read, write, delete";

Policy for Digital Asset Manager —

permission java.io.FilePermission "${dam.content.xfer}/**", "read, write, delete";
permission java.io.FilePermission "${dam.content.xfer}", "read, write, delete";

Policy for Content Intelligence Services —

permission java.io.FilePermission "${cis.content.xfer}/**", "read, write, delete";
permission java.io.FilePermission "${cis.content.xfer}", "read, write, delete";

Preparing Oracle Application Server

Please refer to your product release notes to determine whether Oracle Application Server is a supported application server for your product.

You must disable tag reuse for the application server.

To disable tag pooling for the application:

1. Open orion-web.xml.
2. Change the default value of the <init-param> tags_reuse_default from compiletime to the value none as shown in the following example:

   <servlet>
   <servlet-name>jsp</servlet-name>
   <servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
   <init-param>
   <param-name>tags_reuse_default</param-name>
   <param-value>none</param-value>
   </init-param>
   </servlet>
Preparing the Application Server Host

To disable tag pooling for all applications:
1. Open global-web-application.xml in <ORACLE_HOME>/j2ee/home/config
2. Add the following init-param in the <servlet> element:

   ```xml
   <servlet>
   <init-param>
     <param-name>tags_reuse_default</param-name>
     <param-value>none</param-value>
   </init-param>
   </servlet>
   ```

Preparing Sun Java System Application Server

Please refer to your product release notes to determine whether Sun Java System Application Server is a supported application server for your product.

The following topics describe preparations to deploy a WDK-based application.

Turning off tag pooling

You must turn off tag pooling for the domain in which you deploy WDK-based applications. For the domain in which you will deploy the application, open default-web.xml, for example, install path/domains/domain1/config/default-web.xml. Add the <init-param> to the jsp servlet declaration as follows:

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

Modifying the Sun policy file

You must modify the server.policy file located in the server instance /config directory to add permissions for DFC location variables. Add the following policies if they are not already present in your file:

```text
grant
{
  permission java.util.PropertyPermission "+", "read,write";
  permission java.io.FilePermission "+user.home/-", "read, write, delete";
  permission java.io.FilePermission "+tmp/-", "read, write, delete";
  permission java.io.FilePermission "+java.io.tmpdir/-", "read, write,
```
Preparing the Application Server Host

delete;
permission java.io.FilePermission "${instance.config.dir}/-", "read, write,
delete";
permission java.lang.RuntimePermission "createClassLoader";
permission java.net.SocketPermission ",", "connect,accept";
permission java.lang.RuntimePermission "getProtectionDomain";
permission java.lang.RuntimePermission "shutdownHooks";
permission java.lang.reflect.ReflectPermission ";";
permission java.security.AllPermission;
);

Replace the following variable in these policies or create an environment variable for it so that it can be resolved:

- \$(instance.config.dir): The instance configuration directory, example: /var/opt/SUNWappserver7/domains/domain1/server1/config/

Turning off failover

The Sun application server does not support failover. You must turn it off in the app.xml file located in the custom directory. Add the following lines to custom/app.xml:

```xml
<failover>
  <filter clientenv='portal'>
    <enabled>false</enabled>
  </filter>
  <filter clientenv='not portal'>
    <enabled>false</enabled>
  </filter>
</failover>
```

Preparing to use an external web server

Please refer to your product release notes to determine whether external web servers are supported for your product.

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

UCF content transfer uses chunked transfer encoding, a standard of the HTTP 1.1 specification. Many proxy web servers such as the Sun server implement chunked transfer encoding in a way that does not work properly with UCF. If the external server does not support HTTP 1.1 chunked encoding, you must configure UCF in the WDK-based application to use an alternative chunked encoding. The 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 6

Migrating a WDK-Based Application

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

Table 1, page 37 shows the files, directories, and subdirectories on the application server host that should be backed up.

Table 1. 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>

After installing the new version of the product, copy your backed-up customizations into the new version. Recompile your custom classes to ensure that the custom code still works. For information about migration, refer to Documentum 6.5 System Upgrade and Migration Guide.
Chapter 7

Deploying a WDK-Based Application

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

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

The following topics describe how to deploy the application:

- Preparing the WAR file for deployment, page 39
- Enabling DFC connections to repositories, page 40
- Enabling DFC memory optimization, page 41
- Configuring UCE, page 42
- Enabling presets and preferences repositories, page 43
- Enabling retention of folder structure and objects on export, page 43
- Enabling modal pop-up windows, page 43
- Enable the EMCMF format in WDK-based applications, page 44
- Enabling external searches, page 45
- Deploying multiple applications, page 46

Preparing the WAR file for deployment

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

To deploy a WDK-based application:

1. Download the WDK application WAR file from the EMC download site 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 40. You must perform this procedure before attempting to connect to Documentum repositories.

3. Enable the optional presets and preferences repositories in dfc.properties following the instructions in Enabling presets and preferences repositories, page 43.
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. Refer to the WDK Development Guide for details.
7. Re-archive the WAR file.
8. 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 6.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 Content Server Installation Guide contains information about enabling a repository as a global registry.

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

**To use the dfc.properties file information from the global registry Content Server 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. Open the file dfc.properties that is located in the config subdirectory.
3. Copy the following keys and their values from the file:
   ```
   dfc.docbroker.host[0]=address
   dfc.globalregistry.repository=repository_name
   dfc.globalregistry.username=username
   dfc.globalregistry.password=encrypted_password
   dfc.docbroker.port[0]=port_number
   ```
4. Unpack the application WAR file.
5. Open the dfc.properties file located in WEB-INF/classes within this expanded WAR file directory.
6. Paste in the values that you copied from the global registry dfc.properties.
7. Use a text editor to configure additional properties in this file or make any changes to it.
8. Save the dfc.properties file and deploy the application.

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

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

1. Unpack the application WAR file.
2. Open the file dfc.properties in WEB-INF/classes.
3. 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**

The DFC diagnostics are set to true by default. To free up memory resources, set

\[dfc.diagnostics.resources.enable=false\]

Refer to **Enabling DFC connections to repositories**, page 40, for the procedure of unpacking the war file and modifying dfc.properties. Add the following line to your dfc.properties file:
Configuring UCF

The 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 JRE using UCF, page 42
- 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.

How to configure JRE using UCF

Prior to 6.5 SP1 release, in a client / target workstation, if the installed JRE version is higher than the configured version, you cannot install the configured JRE version. In a portal environment, you may have issues with the installed higher JRE version.

With the 6.5 SP1 release, you can configure Webtop in such a way that the configured JRE version is always installed.

The value of enforceJreInstallation is set to false by default. Below is the snippet from the ucf.installer.config.xml file:

```xml
<platform os="windows" arch="x86">
  <java version="1.5.0" href="win-jre1.5.0_06.zip" exePath="/jre1.5.0_06\bin\java.exe" enforceJreInstallation="false"/>
</platform>
```

If the value of enforceJreInstallation is set to false, then increment the version in the app id element of the ucf.installer.config.xml file to force UCF to pick up the changes made to the ucf.installer.config.xml file.

```xml
<app id="shared" version="6.5.0.036" compatibilityVersion="5.3.0"/>
```

to

```xml
<app id="shared" version="6.5.0.036a" compatibilityVersion="5.3.0"/>
```

For UCF to install always the configured JRE version, set the value of enforceJreInstallation parameter in the ucf.installer.config.xml file to true and restart the application server in the 6.5 SP1 release.

```xml
<platform os="windows" arch="x86">
  <java version="1.5.0" href="win-jre1.5.0_06.zip" exePath="/jre1.5.0_06\bin\java.exe" enforceJreInstallation="true"/>
</platform>
```

**Note:** You can locate the ucf.installer.config.xml file, and the JRE in the following location:

```xml
<app>/wdk/contentXfer
```
Enabling presets and preferences repositories

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

Add your preferences repository settings to app.xml in the /custom directory of the application. Copy the entire <preferencesrepository> element from /wdk/app.xml into /custom/app.xml and then specify your repository. For information on other preferences settings in app.xml, refer to Web Development Kit Development Guide.

Table 2. Preferences configuration elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;preferencesrepository&gt;</td>
<td>Contains a &lt;repository&gt; element. If this element is not present, user preferences are stored in the global repository, which can slow down performance.</td>
</tr>
<tr>
<td>.&lt;repository_path&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>.&lt;repository&gt;</td>
<td>Specifies the repository in which to store preferences, preferably not the global repository.</td>
</tr>
</tbody>
</table>

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

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>
```

The default is false.

Enabling modal pop-up windows

The modal pop-up window is a secondary browser pop-up window with no browser controls either to maximize or minimize the window. This pop-up window appears centered in the screen. The pop-up window provides a similar experience on the web as in desktop, where you can interact
Deploying a WDK-Based Application

with a component in a pop-up window. The user interface for the component appears in a pop-up window (child window) on top of the parent window. If you invoke another component from the child window, the user interface of the component appears on top of the child window and thus stacked one over the other pop-up windows. You cannot access the parent window until you close all the pop-up windows.

The modal pop-up window is supported only on Internet Explorer browser environment. The pop-up window is not 508–compliant and hence it is not supported when 508 accessibility features are turned on through the User Preferences.

In the wdk/app.xml file, <modalpopup> enables and disables the modal pop-up feature.

Table 3, page 44 describes the elements that configure modal windows in app.xml:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;enabled&gt;</td>
<td>Turns on or off modal windows in the application. Valid values: true</td>
</tr>
<tr>
<td>&lt;actionInvocationPostprocessors&gt;</td>
<td>List of action invocation post processors specified in &lt;postprocessor&gt; elements.</td>
</tr>
<tr>
<td>&lt;postprocessor&gt;</td>
<td>Specifies a post processor. The syntax is:</td>
</tr>
<tr>
<td></td>
<td>&lt;postprocessor id=&quot;uniqueId&quot; action=&quot;yourAction&quot; class= &quot;YourActionInvocationPostProcessor&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>where uniqueId is an application-wide unique string identifier for the post processor; yourAction (optional) is the name of the action; and YourActionInvocationPostProcessor is the post processor’s fully qualified Java class.</td>
</tr>
</tbody>
</table>

Enable the EMCMF format in WDK-based applications

Webtop 6.5 SP1 can handle (import/export) EMCMF format email message (dm_message_archive object). Enabling this feature increases the UCF initialization time, so the first content transfer operation may take some time.

To enable import and export of email messages in the EMCMF format:

To enable import and export of email messages in the EMCMF format, do the following:

1. Install DocApps on 5.3 and 6.0 Repositories:
   - On 5.3 and 6.0 content servers, install the DCO, Collaborative Services, and Messaging application (which is part of Email Archive) Docapps.
**Note:** The Webtop 6.5 SP1 email functionality will work with 5.3 or 6.0 Docbase only if you have DCO, Collaboration Services and Messaging App DocApps installed on this Docbase, these DocApps are available with DCO license. If you do not have DCO license, the Webtop 6.5 SP1 email functionality would not be supported on these Docbase.

2. Add the following lines to app.xml (located in the custom directory).

**Note:** Refer to Web Development Kit Development Guide for details on these configuration elements.

```xml
<messageArchivesupport>
  <enabled>true</enabled>
  <supportedfileformat>
    <fileformat>msg</fileformat>
  </supportedfileformat>
  <defaultattachmentobjecttype>dm_document</defaultattachmentobjecttype>
  <storeemfobjectasarchive>false</storeemfobjectasarchive>
  <skipduplicatemessages>true</skipduplicatemessages>
</messageArchivesupport>
```

3. Uncomment the ES1_MRE line in the wdk/contentXfer/ucf.installer.config.xml file.

**Note:** If the EMCMF is enabled after Webtop is being used in the production, incrementing the version in app id element of the ucf.installer.config.xml file is imminent. It forces UCF to pick up the changes made to the ucf.installer.config.xml file as shown below. Otherwise, the ES1_MRE will not get downloaded on to the client machine; it should be installed manually on each client machine.

```xml
<app id="shared" version="6.5.0.036" compatibilityVersion="5.3.0"/>
```

To

```xml
<app id="shared" version="6.5.0.036a" compatibilityVersion="5.3.0"/>
```

**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. The Federated Search documentation provides more information about how to configure the Federated Search server itself.

**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:
   ```java
   dfc.search.ecis.enable=true
   ```
4. Specify the RMI Registry host for the Federated Search server by setting the following:
   
   \[
   \text{dfc.search.ecis.host}=\text{host}_\text{IP} \\
   \text{dfc.search.ecis.port}=\text{port}
   \]
   
   Where
   
   - \text{host}_\text{IP} is IP address or machine name of the Federated Search server.
   - \text{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 \text{dfc.properties} file:

- \text{dfc.search.ecis.backup.host} : host of the backup Federated Search server. Default value is: localhost.
- \text{dfc.search.ecis.backup.port} : port of the backup Federated Search server. Default value is: 3005.
- \text{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.

### Deploying multiple applications

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

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

- Configuring IBM WebSphere after deployment, page 47
- Configuring Oracle Application Server, page 48
- Deploying default virtual link support, page 48
- Accessing the application, page 49
- Testing WDK samples, page 49

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 application class loader first, then click Save.

Set the JSP compiler option to useJDKCompiler to true and the source level to 1.5 (JRE 5) in the configuration file ibm-web-ext.xmi under the application deployment directory, for example:

```
WAS_INSTALL/AppServer/profiles/AppSrv01/config/cells/host_name/
Node01Cell/applications/webtop_war.ear/deployments/webtop/webtop_war/
webtop.war/WEB-INF/ibm-web-ext.xmi
```

Configure the settings as follows:

```xml
<jspAttributes xmi:id="JSPAttribute_1178213473751"
             name="jdkSourceLevel" value="15"/>
<jspAttributes xmi:id="JSPAttribute_3" name="useJDKCompiler"
              value="true"/>
```
Setting com.ibm.ws.webcontainer.invokefilterscompatibility to true

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.

Configuring Oracle Application Server

For the Oracle Application Server, comment out the following lines in webtop\WEB-INF\web.xml:

```
<init-param>
  <param-name>wdk_cache_control_redirect_includetopages</param-name>
  <param-value>!<![CDATA[(\jar)]]></param-value>
</init-param>
```

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>
```

For more information on virtual links, refer to the Web Development Kit Development Guide.

## 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`
Completing the Deployment

Where:
- 
  • *host_name* is the host where the application server is installed
  • *port_number* is the port where the application server listens for connections
  • *virtual_directory* is the virtual directory for the application

A login dialog box appears.

3. Log in to a test repository.
   The login dialog box reappears with the status message **Login Successful**.

4. Type this URL:
   http://*host_name*:port_number/virtual_dir/wdk/samples/index.jsp
   This page displays a list of the available samples.

5. Click **Session Zoo** and type a valid repository username, password, repository name, and domain (if required), then click **Create Connection**.
   The repository is listed in the **All Connected Repositories** section of the page, and the Status message line starts with Successfully connected to repository *repository_name*

6. Continue to experiment with other samples, especially Menu Zoo, Tree Control, and FX Control Pens.
   Some samples have **Create Test Cab** and **Destroy Test Cab** buttons. These create and delete a test cabinet in the repository and require Create Cabinet privileges.
TaskSpace supports 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 at least 8 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. (Refer to the security server documentation.)

2. Configure the web application server to use an external HTTP Server supported by the security server. (Refer to the RSA security server documentation.)

3. Configure the Content Server plugin. (Refer to the Documentum Content Server documentation.)

4. Configure the WDK-based application in app.xml as described in To configure app.xml for a security server single sign-on, page 51.

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/ forment/session. The SSO authentication scheme classes. Modify the properties file to make your preferred SSO authentication scheme (SSOAuthenticationScheme or RSASSOAuthenticatingScheme) 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_CTREMOTE_USER</user_header>
  </sso_config>
</authentication>
```

Table 4, page 52 describes valid values for each element.

**Table 4. Authentication elements (<authentication>)**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;docbase&gt;</code></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><code>&lt;domain&gt;</code></td>
<td>Specifies Windows network domain name.</td>
</tr>
<tr>
<td><code>&lt;service_class&gt;</code></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><code>&lt;sso_config&gt;</code></td>
<td>Contains SSO authentication configuration elements.</td>
</tr>
<tr>
<td><code>&lt;sso_config&gt;</code>, <code>&lt;ecs_plug_in&gt;</code></td>
<td>Specifies name of the Content Server authentication plugin (not the authentication scheme name). Valid value: dm_rsa</td>
</tr>
</tbody>
</table>

52 EMC Documentum Web Development Kit and Webtop 6.5 SP1 Deployment Guide
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;sso_config&gt;</code>.&lt;br&gt;<code>&lt;ticket_cookie&gt;</code></td>
<td>Specifies name of vendor-specific cookie that holds the sign-on ticket. Valid value: CTSESSION</td>
</tr>
<tr>
<td><code>&lt;sso_config&gt;</code>.&lt;br&gt;<code>&lt;user_header&gt;</code></td>
<td>Specifies name of vendor-specific header that holds the username. Valid value: HTTP_CT_REMOTE_USER.</td>
</tr>
</tbody>
</table>
The following topics describe the two methods of installing Application Connectors:

- Overview, page 55
- Enabling installation on Windows 2003, page 56
- GUI installation of Application Connectors, page 56
- Command-line installation of Application Connectors, page 58
- Location of installed files on the client host, page 59

Overview

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

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

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

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

Application Connectors work with UCF content transfer only.

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

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

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

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

The Application Connectors installer will upgrade an existing installation on the client. No uninstall of the previous version is necessary. Application Connectors version 6.x is required in order to connect to Webtop version 6.x. It can also connect to Webtop applications of version 5.3.x.

**Enabling installation on Windows 2003**

The default user rights settings on Windows 2003 prevent Application Connectors installation. Perform the following steps to enable Application Connectors installation:

1. Open gpedit.msc to configure the machine policy by navigating in the Windows Start menu to **Start > Run**.
2. Type gpedit.msc
3. In the left navigation pane, navigate to **Computer Configuration > Administrative Templates > Windows Components > Windows Installer**.
4. Double-click **Disable Windows Installer**, choose **Enabled**, and then choose **Never** in the drop-down list.
5. Log on as a user in the machine Users group to install Application Connectors.

**GUI installation of Application Connectors**

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

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

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

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

5. Click **Install**.

6. Click **Save** and download the file to your desktop.

7. Double-click the saved file to begin installation.
   A welcome screen appears with a warning that the installer disables Desktop Client if it is found.

8. Click **Next**.
   The Customer Information Dialog appears.

9. Choose **Current User** or **All Users**.

10. Click **Next**.

11. For **Enter URL**, type the URL to the Webtop-based installation that you will use within the Office applications, for example:
    
    http://ple legion:8080/webtop

12. Click **Next**.

13. Click **Install** to launch the installer.
   After installation has completed, the Documentum menu is available within the authoring application for which an Application Connector was installed. Figure 2, page 58 shows the Documentum menu within Microsoft Word.
Figure 2. Documentum menu in authoring application

![Documentum menu in authoring application](image)

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

## Command-line installation of Application Connectors

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

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

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

`Documentum-AppConnectors-Client.exe /v"WEBTOPURL=http://server/folder"`
Running the installer in silent mode — The following syntax launches the installer silently from the command line:

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

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

```
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"
```

Location of installed files on the client host

The installer places files in the following locations on the client host. The variable %PROGRAMFILES% is the path to the Program Files directory on the client machine.

Table 5. Location of files installed by Application Connectors on the client host

<table>
<thead>
<tr>
<th>File Type</th>
<th>Location under %PROGRAMFILES%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files used by all Application Connectors</td>
<td>\Documentum\AppConnector</td>
</tr>
<tr>
<td>Files used by a specific Application Connector</td>
<td>Application root directory, for example: \Microsoft Office\OFFICE[10, 11, 12]</td>
</tr>
<tr>
<td>Menu for a Webtop-based application</td>
<td>%PROGRAMFILES%\Microsoft Office\OFFICE[10, 11, 12]\Documentum and subdirectory app_name where app_name must match a value in the app.config files</td>
</tr>
</tbody>
</table>
Documentum Webtop Federated Search Service (FS2) provides the following additional search capabilities for any WDK-based application:

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

The extended search features are available as a licensed option, which requires the installation of DARs in addition to the WDK-based application. The Federated Search server is not required to support Webtop Federated Search Service (FS2). It is only required to search external sources and is installed separately.

**Note:** Although Webtop Federated Search Service (FS2) is a licensed option, no license file is required for the DAR installation.

To enable clustering, use the Webtop Federated Search Service (FS2) installer to deploy the Clustering DAR to a global registry repository. This will also enable search monitoring on every web application that uses the global repository. Deploy the Search Templates DAR on each repository in which you want users to save Search templates.

When you run the installer, it presents a checklist of repositories that are available to the connection broker. The Clustering DAR will be deployed to any global registry repositories version 6.x that you select, and the Search Templates DAR will be deployed to all repositories (version 6.x) that you select.

You can launch the installation on the machine hosting the Content Server or on another machine from which the Content Server is visible. The operating system of the machine from which you launch the installation must be Windows or Linux. Therefore, if the Content Server is on a machine with a Solaris or AIX operating system, you must launch the installation from another machine with a Windows or Linux operating system. The following procedure describes the installation steps.

**Note:** You must have superuser privileges for the repositories in which you want to install the DARs.
Installing the Webtop Federated Search Service (FS2) option

The following procedure describes the installation steps.

**To install the Webtop Federated Search Service (FS2) option**

1. Download the installer file corresponding to your operating system from Powerlink (http://Powerlink.EMC.com). You may also download the ZIP file that contains the language pack(s) for languages other than English. These language pack(s) are necessary to display localized Search templates.

2. Unzip the downloaded file to a temporary directory. The directory should contain:
   - common files for all operating systems:
     - brand.jar
     - composer.jar
     - dars.jar
     - darSetup.jar
   - specific files according to the operating system:
     - dar<Operating_system>SuiteSetup.jar
     - dfc<Operating_system>Setup.jar
     - The installer file: dar<Operating_system>SuiteSetup.exe or dar<Operating_system>SuiteSetup.bin

     where <Operating_system> is the abbreviation or the name of the operating system, such as Win for Windows, or Linux.

3. For Linux operating systems, create the following environment variables, which are required by the installer. If they already exist on the system host, you can skip this step:
   - DOCUMENTUM
     Specifies the full path of the user root directory. Can be any directory in the installation owner’s environment, for example:
     DOCUMENTUM=/export/home/Documentum
   - DOCUMENTUM_SHARED
     Specifies the DFC program root directory. Can be any directory in the installation owner’s environment, for example:
     DOCUMENTUM_SHARED=$DOCUMENTUM

4. Double-click on the installer file that corresponds to your operating system to launch the installation.
   The Welcome screen lists the products that will be installed:
   - DAR Installer version
   - DFC Runtime Environment version

5. Click Next to continue.
6. Click I accept the terms of the license agreement, and then click Next.
The Select Optional Features screen offers the possibility to install optional features for DFC, in this installation, you can install the DFC Developer Documentation (javadocs).

7. Click Next to continue.

8. In the Connection Broker screen, enter a Connection Broker Host Name and Connection Broker Host Port. The connection broker should have access to the repositories on which you want to install the DARs. If the repositories are not visible, you have to run another installation specifying another connection broker. Click Next to continue.
   The Select repositories screen appears.

9. Select the repositories in which the DARs will be deployed. The Clustering DAR will only be deployed in repositories that are configured as global registries. The Search Templates DAR will be deployed in all the repositories that you select.
   Once you select repositories, click Next to continue.
   The Repository Login screen appears.

10. Enter the User name, Password and optional Domain for each repository you previously selected.
    Click Next, and the credentials are tested against the repositories.
    The following screen indicates the directory path where the DARs will be deployed. No files are installed in the local system but an install log is created.

11. Click Next to continue.

12. The DARs are installed. When the installation is complete, click Finish.

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

### Viewing installation log files

Log files are automatically created for the installation of the DARs. To view the installation log files, navigate to the installation directory, and locate the subdirectory dm_log. This directory contains SearchTemplates_installerlog.html, and it contains Clustering_installerlog.html if the clustering feature has been installed.
Chapter 12

Enabling the Webtop Express 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 6, page 65 describes the functionality that is available to Webtop Express users.

Table 6. 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 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.
Chapter 13

Troubleshooting Deployment

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

Wrong JRE used for application server

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

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

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

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 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:

```
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 41 for information on enabling the connection in dfc.properties.

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

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

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 Sun Java plugin installed. You must download and install the Sun 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 69 for troubleshooting information on this problem.

Slow performance

Many performance enhancements are documented in Web Development Guide Development Kit. You can also obtain a system sizing guide from the documentation on Powerlink.

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 28.

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. 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 JBoss and Apache Tomcat, page 29. For the Sun Java System server, refer to Turning off tag pooling, page 33. For Oracle, refer to Preparing Oracle Application Server, page 32.
Caution: After you disable tag pooling, you must clear the cached JSP class files which still may contain pooled tags. Refer to your application server documentation to find the location of the generated class files. For example, Tomcat displays the following error message:

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

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 version 1.5.0_06 of the Sun 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 release notes. 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. Supported JRE versions are listed in the DFC and Webtop application release notes. 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>
   </configuration>
   </defaults>

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. To resolve this, restart the application server.

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
Chapter 14

Deploying a Custom Application

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

The following topic describes how to deploy your custom application:
- Using the comment stripper utility, page 73

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 7, page 73 describes the parameters to use in starting this tool from the console.

<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;!-- and --&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 tool has already been run on some WDK-based applications such as Webtop. The commented files, useful for development, are provided in a JAR file in the base directory: unstripped.jar.
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 8. 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>Release Notes</td>
<td></td>
</tr>
<tr>
<td>Validate your application server and clients operating systems.</td>
<td>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>
## Predeployment Checklist

<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>Web Development Kit Applications Language Pack Installation and Release Notes</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 Chapter 5, Preparing the Application Server Host.</td>
<td></td>
</tr>
</tbody>
</table>
A

Apache Tomcat
  Java heap size, 28
application
  tools, 73
Application Connectors
  architecture, 55
  change menu name, 59
  enabling, 56
  silent install, 59
application server host requirements
  Java heap size, 28
application servers
  performance tuning, 28
  starting, 47
  startup files, 69
  verifying, 47
applications
  multiple, 46

B

backing up customizations, 37
BEA WebLogic
  domains, 29
  Java heap size, 28
  session affinity support, 35
browsers
  Citrix client, 25
  slow display, debugging, 69

C

Citrix client, 25
ClearTrust
  configuration, 51
clients
  preparing, 19
  set JVM, 19
clustered environments, Oracle Application
  Server, 32
comment stripper, 73
configuration, typical, 14
connection
  troubleshooting, 68
connection broker
  troubleshooting, 67
connection brokers, 40
deployment requirement, 13
Content Server
  deployment requirement, 13
  requirements, 15
  versions, 15
Content Server requirements
  global registry, 15
content transfer
  enable in IE7, 20
  enable in Firefox, 21
temporary directory, 15
content transfer operations
  Documentum Application
    Connectors, 55
customizing applications, 9
  backing up customizations, 37
developer licenses, 16

D

DARs, 13
  requirement, 15
deep export
  element. See italic test
default web applications, 48
deploying
  application server host
    requirements, 15
customizing an application, 16
  multiple applications, 16
  planning, 13
  required directories, 15
  single application server, 14
  supporting software, 13
Index

typical configuration, 14
Web Development Kit application, 39
deployment
  completing the process, 47
testing, 49
developer licenses, 16
developing applications, 16
DFC
  global registry, 40
dfc.properties, 41
directories
  content transfer, 15
  permissions, 15
DNS
  requirement, 15
docbroker
troubleshooting, 67
documentation
related, 10
Documentum Administrator
customizing, 16
Documentum Application Connectors
  command-line installation, 58
  content transfer operations, 55
  GUI installation, 56
domains, BEA WebLogic, 29

E
email
  enable EMCMF format in WDK, 44
environment
  variables, 28
extended search DAR
  installing, 61
external web servers, 34

F
Firefox
  preparing for content transfer, 21
forward proxy
  preparation, 34

G
global registry, 40
  requirement, 15
  troubleshooting, 67
global security on IBM WebSphere, 47

I
IBM WebSphere
  global security, 47
  Java heap size, 28
  predeployment requirements, 30
  session affinity support, 35
installation owner
  content transfer directory, 15
  required permissions, 15
installing
  application server software, 27
  DARs, 13, 17
  DocApps, 17
  extended search DAR, 61
  host requirements, 14
  virtual link support, 48
Internet Explorer
  Windows XP SP2, 25
  Internet Explorer 7
    preparing for content transfer, 20

J
Java
  heap size, 28
  memory allocation values, 28
Java heap
  MaxPermSize parameter, 28
JBoss
  predeployment, 29

L
language packs, 16
localization, 16
login page
  troubleshooting, 68

M
MaxPermSize parameter on BEA
  WebLogic, 28
memory
  dfc.properties, 41
menu
  Application Connectors, changing
    name, 59
  modal pop-up windows
    enabling, 43
multiple applications, deploying, 16
O
Oracle Application Server
  clustered environment, 32
  Java heap size, 28
  predeployment requirements, 32
  WebCache, 32
Oracle WebCache, 32
  out of memory errors, 28

P
performance
  DFC setting, 41
  tuning, 28
planning for deployment, 13
policies
  Sun Java System Application Server, 33
  WebSphere, 30
pop-up blockers, 20
predeployment requirements
  IBM WebSphere, 30
  Java heap size, 28
  JBoss, 29
  Oracle Application Server, 32
  Sun Java System Application Server, 33
  Tomcat, 29
  WebLogic domain, 29
preferences
  repository, 43
preinstallation requirements
  application server software, preparing, 27
preparing
  application server host, 27
  client JVM, 19
  clients, 19
presets
  repository, 43
proxy server
  preparation, 34

R
repository
  for presets and preferences, 43
required directories
  content transfer, 15
reverse proxy
  preparation, 34
RSA
configuration, 51

S
security
  WebSphere, 30
  session affinity support, 35
  silent install
    Application Connectors, 59
  single sign-on
    configuration, 51
SSO
  configuration, 51
startup files, application server, 69
Sun Java
  plugin, 19
Sun Java System Application Server
  predeployment requirements, 33

T
tag pooling
  Sun Java System Application Server, 33
  troubleshooting, 69 to 70
Tomcat
  predeployment, 29
tools
  deployment, 73
Trusted Sites, 20
typical configuration, 14

U
UCF content transfer, 19
upgrading
  application server startup files, 69
  overview, 37

V
variables
  environment, 28
viewing WDK samples, 49
virtual link support
  in 5.3 and later installations, 48
  legacy support, 48

W
WAR file
  preparing for deployment, 39
<table>
<thead>
<tr>
<th>WDK</th>
<th>Webtop Express</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable EMCMF format, 44</td>
<td>installing, 65</td>
</tr>
<tr>
<td>WDK applications</td>
<td>Webtop Federated Search Service (FS2)</td>
</tr>
<tr>
<td>accessing, 49</td>
<td>log files, 63</td>
</tr>
<tr>
<td>deploying, 39</td>
<td>Windows</td>
</tr>
<tr>
<td>verifying, 49</td>
<td>XP SP2, 25</td>
</tr>
<tr>
<td>web servers, external, 34</td>
<td></td>
</tr>
</tbody>
</table>