# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Chapter 1</td>
<td>Planning for Deployment</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Required and optional supporting software</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Typical configuration</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Content Server requirements</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Application server host requirements</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Directory name restriction</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Content transfer directory permissions</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>DNS resolution</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Deploying multiple applications</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Deploying language packs</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Customizing an application</td>
<td>15</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Preparing the Client Host</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Ensuring a certified JVM on browser clients</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Enabling HTTP content transfer in Internet Explorer 7</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Enabling UCF content transfer in Internet Explorer 7 on Windows Vista</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Enabling content transfer in Firefox 2</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Clearing the browser cache</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Supporting Outlook mail message archives</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Using Citrix Presentation Server Client</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Turning off the pop-up blocker in Internet Explorer</td>
<td>21</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Preparing the Application Server Host</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Setting the Java memory allocation</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Turning off failover</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Preparing environment variables for non-default DFC locations</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Preparing Apache Tomcat</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Preparing BEA WebLogic</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Supporting large content transfer operations in a managed server environment</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Preparing IBM WebSphere</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Supporting failover in a cluster</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Applying policies for WebSphere global security</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Supporting non-default content transfer locations</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Preparing Oracle Application Server</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Preparing Sun Java System Application Server</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Turning off tag pooling</td>
<td>30</td>
</tr>
</tbody>
</table>
Chapter 4  Preparing to Upgrade a WDK-Based Application ............................................. 33
            Backing up customizations ........................................................... 33
            Cleaning up the application server startup script ............................... 34
            Uninstalling previous versions ....................................................... 34

Chapter 5  Deploying a WDK-Based Application ......................................................... 37
            Preparing the WAR file for deployment ........................................... 37
            Enabling DFC connections to repositories ................................... 38
            Enabling DFC memory optimization ............................................. 39
            Enabling presets and preferences repositories ............................... 40
            Enabling external searches ......................................................... 40
            Deploying multiple applications ................................................... 41

Chapter 6  Completing the Deployment ................................................................. 43
            Configuring UCF .............................................................................. 43
            Configuring IBM WebSphere after deployment ................................ 44
                Changing the classloader and compiler settings ............................ 44
            Configuring single sign-on through security servers ....................... 44
            Deploying default virtual link support ...................................... 47
            Accessing the application .............................................................. 48
            Testing WDK samples ................................................................... 49

Chapter 7  Installing Application Connectors ......................................................... 51
            Overview ......................................................................................... 51
            Enabling installation on Windows 2003 .......................................... 52
            GUI installation of Application Connectors ..................................... 52
            Command-line installation of Application Connectors ...................... 54
            Location of installed files on the client host ...................................... 55

Chapter 8  Installing Webtop Extended Search DocApps ............................................ 57
            Installing the Webtop Extended Search option .................................. 58
            Viewing installation log files .......................................................... 60

Chapter 9  Troubleshooting Deployment ................................................................. 61
            Wrong JRE used for application server ........................................... 61
            No global registry or connection broker ......................................... 61
            No connection to repository ........................................................... 62
            Page not found error in browser ..................................................... 62
            Login page incorrectly displayed .................................................... 62
            Slow performance .......................................................................... 63
            Out of memory errors in console or log .......................................... 63
            Slow browser display ...................................................................... 63
            DFC using the wrong directories on the application server .................. 63
### Table of Contents

Tag pooling problem ................................................................. 64
Citrix client problems ............................................................... 64

**Chapter 10** Deploying a Custom Application ................................. 67
Using the comment stripper utility ............................................... 67
Packaging a WAR file for deployment .......................................... 68

**Chapter 11** Installing Records Manager Administrator .................. 69
Download RMA files .................................................................. 69
Create global repository ......................................................... 70
Create RM repository ............................................................. 71
Install RMA content server files ................................................ 71
  Installing RMA Content Server files on Windows ...................... 71
  Installing RMA Content Server files on Unix/Linux hosts .......... 72
Install docapps ..................................................................... 72
Prepare the application server .................................................. 74
Deploy the RMA war file ......................................................... 74
Modify the dfc.properties file .................................................. 75
Register the DFC client ............................................................ 75

**Chapter 12** Upgrading Records Manager Administrator .................. 77
Upgrading and Migrating RMA 5.3 SP4 or SP5 to RMA 6.0 ............... 78
Upgrading Legacy RM 5.2.5 SP5 to RMA 6.0 ............................... 80

**Chapter 13** Configuring Records Manager Administrator ............... 81

**Appendix A** Pre-Installation Checklist ...................................... 83

**Appendix B** Installing Brava! Webtop Server for RMA .................. 85
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>13</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Documentum menu in authoring application</td>
<td>54</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Supported upgrade paths</td>
<td>77</td>
</tr>
<tr>
<td>Figure 4</td>
<td>RMA Webtop alias</td>
<td>85</td>
</tr>
</tbody>
</table>
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Output directories and JVM custom properties</td>
<td>29</td>
</tr>
<tr>
<td>Table 2</td>
<td>Directories and files to back up</td>
<td>33</td>
</tr>
<tr>
<td>Table 3</td>
<td>Preferences configuration elements</td>
<td>40</td>
</tr>
<tr>
<td>Table 4</td>
<td>Authentication elements (&lt;authentication&gt;)</td>
<td>46</td>
</tr>
<tr>
<td>Table 5</td>
<td>Location of files installed by Application Connectors on the client host</td>
<td>55</td>
</tr>
<tr>
<td>Table 6</td>
<td>Comment stripper utility parameters</td>
<td>67</td>
</tr>
<tr>
<td>Table 7</td>
<td>Preinstallation tasks</td>
<td>83</td>
</tr>
</tbody>
</table>
Preface

This guide provides preliminary procedures for deploying EMC Documentum Records Manager version 6 (RM 6.0) along with the actual procedure needed to complete the installation. The Records Manager client, Records Manager Administrator (RMA), displays the Records Manager (RM), Retention Policy Services (RPS), and Physical Records Manager (PRM) nodes. RPS and PRM functionality however, are disabled until you have them activated with their respective license keys. Each can be activated separately by installing the respective license key during the installation process or at a later time. RMA integrates RM, RPS, and PRM on top of Webtop to provide unified functional access from a single user interface. Only Webtop functionality is accessible if no license keys are installed.

The chapters for installing, upgrading, and configuring are preceded by chapters consisting of preliminary procedures commonly used to plan and prepare the content server and the application server used for a WDK-based application deployment. A pre-installation checklist and Brava! Viewer instructions are also provided.

Refer to the Records Manager Release Notes, before following any installation or upgrade procedure, to verify that your system requirements match or exceed the specifications indicated in the section for the Environment and System Requirements. 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.

RMA is a web application built on WDK and can therefore be customized using WDK. The deployment process is largely the same as for other WDK-based applications. The steps specific to deploying RMA 6.0 start in Chapter 11. For additional information on developing or customizing applications which are WDK-based, refer to the Web Development Kit and Client Applications Development Guide.

Intended Audience

This guide is intended for the system administrator or IT professional who installs enterprise software. Installing Records Manager Administrator requires administrative access to the computers on which the components are installed.
Revision History

The following changes have been made to this document.

**Revision History**

<table>
<thead>
<tr>
<th>Revision Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2007</td>
<td>Initial Publication.</td>
</tr>
<tr>
<td>October 2007</td>
<td>Revised instructions for installing docapps.</td>
</tr>
</tbody>
</table>
Chapter 1

Planning for Deployment

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 11
- Typical configuration, page 12
- Application server host requirements, page 14
- Deploying multiple applications, page 15
- Deploying language packs, page 15
- Customizing an application, page 15

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 dcf.properties file. Refer to To configure connections in dcf.properties before deployment, page 39 for information on configuring the docbroker before deployment.

- J2EE application server or servlet container

All WDK-based applications require DocApps that must be installed in the repository. The Webtop DocApps are provided in Content Server version 6. If your application supports connections to a Content Server version 5.3.x, you must have a Content Server
6 global registry. The following features in the Webtop DocApps will not be available with a 5.3.x Content Server:

- ACS distributed write
- Create relationship
- Presets

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.

**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 13* shows the network components.
Planning for Deployment

Figure 1. Basic WDK host configuration

![Diagram of WDK host configuration](image_url)

**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 WebLogic 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.

**Content Server requirements**

The following topics describe Content Server requirements. If your application is certified for connections to either 5.3 or version 6 Content Server repositories, you must use a Content Server 6 global registry.

If you connect to a 5.3 repository, the following functionality will not be available:

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

Content Server 6 installs the required DocApps for a WDK-based application. You do not need to perform a separate installation of these.

A global registry 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 38.

⚠️ **Caution:** The global registry must be Content Server version 6.

### 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 Unified Client Facilities (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 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 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 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.

⚠️ **Caution:** A 5.3.x application cannot share the same instance as a version 6 application. One or both of the applications will not work properly.

**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 (JavaServer 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 2

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 17
- Enabling HTTP content transfer in Internet Explorer 7, page 18
- Enabling content transfer in Firefox 2, page 19
- Clearing the browser cache, page 20
- Supporting Outlook mail message archives, page 20
- Using Citrix Presentation Server Client, page 20
- Enabling UCF content transfer in Internet Explorer 7 on Windows Vista, page 18

Ensuring a certified JVM on browser clients

Browser client hosts require a certified version of the Sun Java virtual machine (JVM) to initiate content transfer in a WDK application. New machines may not have a VM 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. If the JVM required for UCF is not present on the client machine, UCF uploads to the client a private JVM. This VM does not replace the JVM that is used by the browser. 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 is 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.
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 2

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.
Clearing the browser cache

Browsers cache JavaScript. WDK-based applications contain changes to the JavaScript files that existed in 5.x applications. Cached JavaScript in the browser from a 5.x WDK-based application may cause errors when the user connects to a version 6 application. Clear the browser JavaScript cache to avoid these errors.

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.

The 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 download and install a converter. This converter can be automatically installed as part of the UCF download.

To enable automatic download, uncomment the ExMRE.exe section in the app\wdk\contentXfer\ucf.installer.config.xml file on the application server.

The converter can take a long time to download and install. You can avoid the delay by deploying the ExMRE.exe installer using standard mechanisms such as Microsoft Systems Management Server (SMS).

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\pradeep\Documentum\ucf.
Turning off the pop-up blocker in Internet Explorer

Windows XP SP2 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 3

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. For uninstall procedures, refer to Uninstalling previous versions, page 34.

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 24
  Required information for all application servers
- Turning off failover, page 24
- Preparing environment variables for non-default DFC locations, page 24
  Information for enterprise environments that do not use the default (recommended) DFC environment locations.
- Preparing Apache Tomcat, page 25
- Preparing BEA WebLogic, page 25
- Preparing IBM WebSphere, page 26
- Preparing Oracle Application Server, page 29
- Preparing Sun Java System Application Server, page 30
- Preparing to use an external web server, page 31

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:

- \texttt{Xms1024m}
- \texttt{Xmx1024m}

Application servers can slow down, throw exceptions, or crash with an application that has a large number of Java Server Pages. Set the \texttt{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.

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 \texttt{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 \texttt{app.xml} in the custom directory and add the following element:

\begin{verbatim}
<failover>
  <enabled>false</enabled>
</failover>
\end{verbatim}

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 \texttt{dfc.data.dir}. This location is specified as the value of the key \texttt{dfc.data.dir} in \texttt{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, 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 Apache Tomcat**

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

You must disable tag reuse in Apache Tomcat in the web.xml file of the Tomcat /conf directory. Find the JSP servlet entry in web.xml. Add the enablePooling initialization parameter and set it to false:

```xml
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>enablePooling</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>

<servlet>
  <servlet-name>fork</servlet-name>
  <init-param>
    <param-name>fork</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>

<servlet>
  <servlet-name>xpoweredBy</servlet-name>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>Apache Tomcat</param-value>
  </init-param>
</servlet>

<load-on-startup>3</load-on-startup>
</servlet>
```

**Preparing BEA WebLogic**

Please refer to your product release notes to determine whether BEA WebLogic is a supported application server for your product.
Preparing the Application Server Host

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 global 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 global security

If WebSphere global security is enabled for the application server, 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>`.

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 dfc.properties key dfc.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 dfc.properties keys dfc.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 dfc.registry environment variable must match the location specified in dfc.properties for the key dfc.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";
```
Preparing the Application Server Host

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:

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.

permission java.io.FilePermission "${documentum}/${}/-", "read, write, delete";
permission java.io.FilePermission "${documentum}/*", "read, write, delete";

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";

Supporting non-default content transfer locations

If you want to use non-default locations for content transfer, you must create custom JVM properties using the WebSphere administration console. Create a JVM custom property named dfc.data to reflect the path to the location. The value must match the value of dfc.data.dir in dfc.properties. Table 1, page 29 shows the custom properties you must create each DFC output directory that is not the default location.
Preparing the Application Server Host

Table 1. Output directories and JVM custom properties

<table>
<thead>
<tr>
<th>DFC output directory key</th>
<th>JVM custom property name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfc.data.dir</td>
<td>dfc.data</td>
</tr>
<tr>
<td>dfc.user.dir</td>
<td>dfc.user</td>
</tr>
<tr>
<td>dfc.checkout.dir</td>
<td>dfc.checkout</td>
</tr>
<tr>
<td>dfc.export.dir</td>
<td>dfc.export</td>
</tr>
<tr>
<td>dfc.registry.file</td>
<td>dfc.registry</td>
</tr>
</tbody>
</table>

For dfc.registry, type the path to the parent directory of the file. Refer to the documentation for WebSphere for information on modifying JVM custom properties.

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:

   ```xml
   <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>
   
   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 following <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:

```java
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, 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/

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 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 and Client Applications 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 4

Preparing to Upgrade 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. Review this chapter and perform the tasks described in it before upgrading WDK application.

This chapter contains the following information:

- Backing up customizations, page 33
- Cleaning up the application server startup script, page 34
- Uninstalling previous versions, page 34

Back up customizations

If you customized a 5.x WDK-based application, Table 2, page 33 shows the files, directories, and subdirectories on the application server host that should be backed up. Customization of Documentum Administrator is not supported.

**Table 2. 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 and 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>
</tbody>
</table>
Preparing to Upgrade a WDK-Based Application

<table>
<thead>
<tr>
<th>Directory/file</th>
<th>To back up if present</th>
</tr>
</thead>
<tbody>
<tr>
<td>/custom/src subdirectories</td>
<td>Custom source files</td>
</tr>
<tr>
<td>/WEB-INF/tlds</td>
<td>Custom tag libraries</td>
</tr>
</tbody>
</table>

After upgrading, recompile your custom classes to ensure that the custom code still works. Add your backed up files to the new application for testing. For information about migration, refer to System Migration Guide.

Cleaning up the application server startup script

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. Each WDK-based application contains the libraries required for version 6 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.

Uninstalling previous versions

It is recommended that you uninstall every application that uses DFC 5.x, and DFC 5.x itself, before deploying an application based on WDK 6. The following topics describe how to uninstall previous versions of WDK-based applications. They do not apply to WDK-based application of version 6. Those applications are not installed, so they cannot be uninstalled.

When you run the 5.3.x uninstaller, the application server must be in the following state:

- Apache Tomcat must be stopped.
- BEA WebLogic must be running.
- IBM WebSphere must be running.
- Oracle Application Server must be running.
- Sun Java System Application Server must be running.

You must ensure that the application server is in the correct state before running the uninstaller.
Preparing to Upgrade a WDK-Based Application

Uninstalling a WDK-based application — If only one instance of a product is installed on the application server host, use the Add/Remove Programs feature of Windows to uninstall the software. If multiple instances are installed or you are not sure how many instances are installed, navigate to the uninstaller directory for each instance you are removing and use the following procedure.

To uninstall WDK or a WDK application
1. Connect to the host as the application server installation owner.
2. Stop all other running programs.
3. Navigate to the appropriate uninstaller directory. Perform one of the following:
   a. On Windows:
      
      \DFC\program\root\uninst\product\name\app\server\name\virtual\dir
      
      The default path for the DFC program root on Windows is C:\Program Files\Documentum.
   b. On UNIX:
      
      $\DOCUMENTUM\_SHARED\uninst\product\name\app\server\name\virtual\dir

4. Perform one of the following:
   a. On Windows, double-click uninstall.exe.
   b. On UNIX, type the following command:
      
      ./uninstall.bin

Uninstalling DFC — After all Documentum products are removed from the application server host, Documentum Foundation Classes (DFC) can be uninstalled and all remaining Documentum files, directories, and registry entries (if applicable) can be removed. You can also remove DFC from client hosts that have no Documentum applications installed.

To uninstall DFC on Windows
1. Select Start > Settings > Control Panel > Add or Remove Programs.
2. Select Documentum DFC Runtime Environment.
3. Click Remove. If this product is the last Documentum product on the host, a dialog box is displayed that asks if you want to remove the JDK and other Documentum settings and directories. This removes Documentum from your registry settings.

To uninstall DFC on UNIX or Linux
1. Navigate to $\DOCUMENTUM\_SHARED\_uninst\dfc.
2. Start the uninstaller:
   
   ./uninstall.bin
3. Choose Remove Documentum completely.

**Uninstalling a WDK application that was installed by another user** — To uninstall Documentum products installed by a user other than yourself, you must change two parameters in the uninstall.ini file.

**To modify the uninstall.ini file:**

1. Navigate to the product’s _uninst directory.
2. Open the uninstall.ini file in a text editor.
3. Modify the values of the USER_ACCOUNT and USER_DOMAIN parameters:

   USER_ACCOUNT=new_user_account
   USER_DOMAIN=new_user_domain

4. Save the uninstall.ini file.

**Uninstalling a product that was installed on UNIX as root** — If you installed a product as root on a UNIX host, you cannot uninstall the product using the uninstaller. You must remove all Documentum installer entries from the /var/sadm/pkg directory. These are files with names that begin with IS.

Run the following command to list all files in the directory beginning with IS:

```
ls -rf /var/sadm/pkg/IS*
```

If no non-Documentum files are returned by the list command, remove all of the Documentum installer files with this command:

```
rm -rf /var/sadm/pkg/IS*
```

**Caution:** If there are non-Documentum files that match the IS* wildcard expression, you must enumerate the Documentum files in the rm command.
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 37
• Enabling DFC connections to repositories, page 38
• Enabling presets and preferences repositories, page 40
• Enabling external searches, page 40
• Deploying multiple applications, page 41

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 38. 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 40.
4. Add or migrate customizations from previous WDK-based applications.
5. Apply language packs if you have purchased them.
6. Re-archive the WAR file.
7. 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 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 *System Deployment 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
   ```
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.
   
   ddc.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:
   
   ddc.docbroker.port=port_number

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

6. Add the username of the dm_bof_registry user to the following key:
   
   ddc.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:
   
   ddc.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 ddc.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 in dfc.properties. Refer to Enabling DFC connections to repositories, page 38 for the procedure of unpacking the war file and modifying dfc.properties. Add the following line to your dfc.properties file:
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 3. 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 external searches

To allow users to search external sources, an administrator must configure a connection to an ECIS search server. (The ECI 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.
To configure the connection to an ECIS search server

1. Unpack the client application WAR file.
2. Open the file dfc.properties in WEB-INF/classes.
3. Enable the ECIS search server by setting the following:
   
   `dfc.search.ecis.enable=true`

4. Specify the RMI Registry host for the ECI Server by setting the following:
   
   `dfc.search.ecis.host=host_IP`
   `dfc.search.ecis.port=port`

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

Deploying multiple applications

A WDK-based application of version 5.3.x cannot share the same application server instance as a version 6 application. Two or more WDK-based applications of version 6 can share the same application server instance if they are version 6 or higher.
Chapter 6

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 UCF, page 43
- Configuring IBM WebSphere after deployment, page 44
- Configuring single sign-on through security servers, page 44
- Deploying default virtual link support, page 47
- Accessing the application, page 48
- Testing WDK samples, page 49

Configuring UCF

The *Web Development Kit and Client Applications Development Guide* contains the following procedures:

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

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

Configuring IBM WebSphere after deployment

To complete the deployment, perform the following procedures.

Changing the classloader and compiler settings

Change the classloader setting for the WDK-based application module in WebSphere, in the Manage Modules section of the administration console. Select the WAR file and for Classloader order choose Classes loaded with 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:

  <jspAttributes xmi:id="JSPAttribute_1178213473751" name="jdkSourceLevel" value="15"/>
  <jspAttributes xmi:id="JSPAttribute_3" name="useJDKCompiler" value="true"/>

This step is required because the default compiler assumes the presence of IBM WebSphere Studio.

Configuring single sign-on through security servers

Refer to your product release notes to determine whether the product supports single sign-on.

Content Server supports pluggable authentication or single sign-on (SSO) using RSA ClearTrust or CA SiteMinder.

RSA ClearTrust users must have the same login names as the Content Server repository. User names are case-sensitive for Server, so ClearTrust user names must have the same case as the repository login. Errors in authentication are logged in the /Documentum/dba/log/dm_rsa.log file.

For CA SiteMinder, you must set up a SiteMinder realm to perform authentication for WDK applications. The dm_netegrity plugin installed in the Content Server decodes
the SMSESSION token sent from WDK for authentication. The plugin contacts the CA server to verify that the token is valid. Errors in authentication are logged in the /Documentum/dba/log/dm_netegrity.log file.

Perform the following procedure to enable single sign-on in a WDK-based application.

**To enable single sign-on (SSO)**

1. Configure the RSA ClearTrust or CA SiteMinder 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 or CA 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 45.
5. RSA only: Create a directory named rsaConfig under the root WDK-based application directory. Copy into this directory two files: aserver.conf copied from the ClearTrust server and webagent.conf copied from RSA web agent.
   If you make changes to the original files, you must copy the changed files to your WDK-based application rsaConfig directory. For more information on these files, refer to the RSA documentation.
6. Locate the file AuthenticationScheme.properties in WEB-INF/classes/com/documentum/web/formext/session. The single sign-on (SSO) authentication scheme classes. Modify the properties file to make your preferred SSO authentication scheme (SSOAthenticationScheme or RSASSOAuthenticalScheme) first in the list of authentications that are attempted during login.
   If the Docbase Login scheme is listed before the SSO scheme, the user will be presented with a login screen instead of single sign-on.
7. Restart the application server.

The WDK SSO Authentication Scheme for CA SiteMinder needs three pieces of information in order to authenticate an HTTP session against a repository:

- Name of the Authentication Plugin that is used in the content server.
- Name of the ticket that will be retrieved from a vendor-specific cookie.
- Username, which is retrieved from a vendor-specific HTTP requests header or remote user.

**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 element <sso> under the existing <authentication> element similar to the following example. This example is for RSA. Replace the repository name in the <docbase> element. If you are configuring the application to use CA SiteMinder, use the appropriate values specified in Table 4, page 46.

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

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;docbase&gt;</td>
<td>Specifies default repository name. When SSO authentication is enabled but a repository name is not explicitly spelled out by the user nor defined in this element, the sso_login component is called. In this case the component will prompt the user for the repository name.</td>
</tr>
<tr>
<td>&lt;domain&gt;</td>
<td>Specifies Windows network domain name</td>
</tr>
<tr>
<td>&lt;service_class&gt;</td>
<td>Specifies fully qualified name of class that provides authentication service. This class can perform pre- or post-processing of authentication.</td>
</tr>
<tr>
<td>&lt;sso_config&gt;</td>
<td>Contains single sign-on authentication configuration elements</td>
</tr>
<tr>
<td>&lt;ecsPlug_in&gt;</td>
<td>Specifies name of the Content Server authentication plugin (not the authentication scheme name). Valid values: RSA: dm_rsa CA: dm_netegrity</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>&lt;sso_config&gt;.</td>
<td>Specifies name of vendor-specific cookie that holds the sign-on ticket. Valid values: RSA: CTSESSION CA: SMSESSION</td>
</tr>
<tr>
<td>&lt;ticket_cookie&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;sso_config&gt;.</td>
<td>Specifies name of vendor-specific header that holds the username. Valid values: RSA: HTTP_CT_REMOTE_USER. CA: The user_header value is dependent on the settings in the webagent configuration object in the policy server. The default is either SMUSER or SM_USER depending on whether the flag &quot;LegacyVariable&quot; is set to true or false. If false, use SMUSER, if true, use SM_USER.</td>
</tr>
<tr>
<td>&lt;user_header&gt;</td>
<td></td>
</tr>
</tbody>
</table>

### 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/repoistory-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:

```xml
<context-root>/</context-root>
```

For more information on virtual links, refer to the *WDK and Client Applications 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 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

   ```text
   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:
   
   \[ http://host_name:port_number/virtual_directory/component/login \]
   
   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.
Chapter 7

Installing Application Connectors

The following topics describe the two methods of installing Application Connectors:

- Overview, page 51
- Enabling installation on Windows 2003, page 52
- GUI installation of Application Connectors, page 52
- Command-line installation of Application Connectors, page 54

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:
Installing Application Connectors

- 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 is required in order to connect to Webtop version 6. 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 gedit.msc to configure the machine policy by navigating in the Windows Start menu to Start > Run.
2. Type gedit.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 Webto-based p installation that you will use within the Office applications, for example:
    **http://plelegion: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 **54** shows the Documentum menu within Microsoft Word.
**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
Installing Application Connectors

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.

Using msiexec — Installing to a specific Microsoft application:

`msiexec /i "\\server\folder\Documentum-AppConnectors-Client.exe" /q ALLUSERS=1 INSTALL="word"

Uninstallation — Command-line uninstallation uses the following syntax:

`msiexec /q /x "\\server\folder\Documentum-AppConnectors-Client.exe"

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

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

Silent mode:

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

Client for Outlook — For Documentum Client for Outlook, client side:

`Documentum-Client-for-Outlook.exe /v"WDK_SERVER_URL=http://server/folder"

Silent mode:

`Documentum-Client-for-Outlook.exe /s /v"/qn WDK_SERVER_URL=http://server/folder"

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>File Type</td>
<td>Location under %PROGRAMFILES%</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Files used by a specific Application Connector</td>
<td>Application root directory, for example: \Microsoft Office\OFFICE{10, 11}</td>
</tr>
<tr>
<td>Menu for a Webtop-based application</td>
<td>%PROGRAMFILES%\Microsoft Office\OFFICE{10, 11}\Documentum and subdirectory app_name where app_name must match a value in the app.config files</td>
</tr>
</tbody>
</table>
Webtop Extended Search DocApps allow users to benefit from additional search features:

- **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.

**Note:** Extended Search does not require a license key or ECI Services server. The ECI Services server is only required to search external sources.

The DocApps installer must deploy the DocApps to a global registry repository in order to enable clustering. This will enable search monitoring on every web application that uses the global repository. The Search templates must be deployed on each repository in which you want users to save Search templates.

When you run the installer, it will present a checklist of repositories that are available to the connection broker. The Clustering DocApp will be deployed to any global registry repositories version 6 that you select, and the Search Templates DocApp will be deployed to all repositories (version 5.3.x or 6) 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 DocApps.
Installing the Webtop Extended Search option

To install the Webtop Extended Search option

1. Download the installer file corresponding to your operating system from Powerlink(https://Powerlink.EMC.com):
   - For Windows: WebtopExtendedSearch60-win.zip
   - For Solaris: WebtopExtendedSearch60-sol.tar
   - For Linux: WebtopExtendedSearch60-lin.tar

2. Unzip the downloaded file to a temporary directory. The directory should contain:
   - For Windows:
     - ExtendedSearchDocAppWinSuiteSetup.jar
     - ExtendedSearchDocAppSetup.jar
     - dfcWinSetup.jar
     - docApps.jar
     - The installer file: WebtopExtendedSearch-win.exe
   - For Solaris:
     - ExtendedSearchDocAppSolSuiteSetup.jar
     - ExtendedSearchDocAppSetup.jar
     - dfcSolSetup.jar
     - docApps.jar
     - The installer file: WebtopExtendedSearch-sol.bin
   - For AIX:
     - ExtendedSearchDocAppAixSuiteSetup.jar
     - ExtendedSearchDocAppSetup.jar
     - dfcAixSetup.jar
     - docApps.jar
     - The installer file: WebtopExtendedSearch-aix.bin
Installing Webtop Extended Search DocApps

- For Linux:
  - ExtendedSearchDocAppLinuxSuiteSetup.jar
  - ExtendedSearchDocAppSetup.jar
  - dfcLinuxSetup.jar
  - docApps.jar
  - The installer file: WebtopExtendedSearch-lin.bin

3. 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:
   - Webtop Extended Search DocApps version
   - DFC Runtime Environment version

4. Click Next to continue.

   The License agreement screen appears.

5. 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:
   - DFC Developer Documentation (javadocs)
   - Primary Interop Assembly Installer

6. Click Next to continue.

7. In the Connection Broker screen, enter a Connection Broker Host Name and Connection Broker Host Port and click Next.

   The Select repositories screen appears.

8. Select the repositories in which the DocApps will be deployed. The Clustering DocApp will only be deployed in repositories that are configured as global registries. The Search Templates DocApp will be deployed in all the repositories that you select.

   Note: The search templates are installed only in the languages already activated for the repository. For example, installing the Search templates option in a repository where only the German, English, and French languages are activated installs only the Search templates localized in German, English, and French. When another language is activated in the repository after the installation of the Search templates option, the Search templates are not localized for this language. You need to reinstall the Search templates DocApp to have the Search templates localized for this newly activated language.

   Once you select repositories, click Next to continue.

   The Repository Connection screen appears.
9. 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 DocApps will be deployed. No files are installed in the local system but an install log is created.

10. Click Next to continue.

11. DFC is installed first in each repository, and then the DocApps. When the installation is complete, click Finish.

**Viewing installation log files**

Log files are automatically created for the installation of the DocApps. 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.
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].[/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 and 
to the System Deployment Guide for system-level information about the 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:

at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:432)
Caused by: DfDocbrokerException:: THREAD: main; MSG: [DFC_DOCBROKER_REQUEST_FAILED] 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 39 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

Page not found error in browser

If the client browser uses the Java 1.6 JRE, the application URL may return a "page cannot 
be found". Check your product release notes for the supported browser environments.

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 64 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 24.

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

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

**Slow browser display**

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.
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 Apache Tomcat, page 25. For the Sun Java System server, refer to Turning off tag pooling, page 30. For Oracle, refer to Preparing Oracle Application Server, page 29.

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

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

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\Pradeep\Documentum.
2. Edit ucf.installer.config.xml in /wdk/contentXfer in the WDK application. Change every environment variable in this file that uses the Java home directory $java{user.home} to use the roaming profile environment variable:

```xml
<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>
  </ucfInstallsHome>
</defaults>
```

3. Save and restart the application server.
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.

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. This utility is called by the WAR file tool CreateInstallerWAR, so you do not need to use the comment stripper if you are using CreateInstallerWAR. Table 6, page 67 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;!....-&gt; and &lt;!....--&gt;</td>
</tr>
<tr>
<td>j</td>
<td>Removes JSP and JavaScript / * ... * / comments</td>
</tr>
<tr>
<td>r</td>
<td>Recurses directories from current</td>
</tr>
</tbody>
</table>
Deploying a Custom Application

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

Packaging a WAR file for deployment

You can use the tool CreateInstallerWAR to package your web application for deployment.

The CreateInstallerWAR tool strips out all comments from files with the following extensions: .html, .htm, .js, and .jsp. This improves the performance of your web application.

**To create a WDK-based WAR file**

1. Ensure that you have the Sun Java SDK root directory on your system path.
2. Include \WEB-INF\classes on your classpath:
   ```
   set classpath  App_root_directory\WEB-INF\classes
   where App_root_directory is the root directory for your custom Web application.
   ```
3. From the command prompt, type the following command on a single line:
   ```
   java com.documentum.web.tools.CreateInstallerWAR
   source_virtual_directory
   destination_file
   ```
   where `source_virtual_directory` represents the root directory for your custom Web application and `destination_file` represents a name for your new WAR file.
4. Remove the entry in your system classpath that you created in step 2, in order to run the WDK application.
Chapter 11

Installing Records Manager Administrator

Note: Use these procedures only if you are installing Records Manager Administrator (RMA 6.0) for the first time. Refer to the next chapter for the upgrade procedures if Records Manager is already installed.

Refer to the Records Manager Release Notes, 6.0 before starting the installation process, to verify that your system requirements match or exceed the specifications indicated in the section for the Environment and System Requirements.

Each section listed here represents a step in the installation process. You can see at a glance all the major steps involved. Refer to each section in the order provided to complete the installation.

1. Download RMA files, page 69
2. Create global repository, page 70
3. Create RM repository, page 71
4. Install RMA content server files, page 71
5. Install docapps, page 72
6. Prepare the application server, page 74
7. Deploy the RMA war file, page 74
8. Modify the dfc.properties file, page 75
9. Register the DFC client, page 75

Download RMA files

Download the RMA docapps, the RMA application war file, and the Content Server files to the download folder.

You will need the license keys for RPS and RM. If you are installing PRM, there is also a separate license key required. The license keys are identified as .txt files.

- rpsdocapp.tar
- rmdocapp.tar
Create global repository

A global repository is a mandatory requirement for the RMA application.

Create a global repository on the Content Server, for shared access to privileged code (privileged DFC), if one has not already been created, and install both the forms docapp
(Forms.zip) and the forms adaptor docapp (rm_forms_adaptor_docapp.tar) using DAI 5.3.5.

**Note:** Both the forms and the forms adapter docapps need to be installed on the global repository. The forms docapp must be installed ahead of the forms adaptor docapp. Only the forms docapp (Forms.zip) needs to be installed on each individual RM repository (application repository) and the global repository.

## Create RM repository

Create an application repository for RMA (RM repository). During the creation of the repository, you can enter the license keys for RPS, RM, and PRM. Note that you can also enter the keys after the repository is created by using the Documentum Administrator.

## Install RMA content server files

Install the Content Server files according to the instructions pertaining to the applicable platform, Windows or Unix.

### Installing RMA Content Server files on Windows

To complete the RMA installation, the RMA Server Files for the Content Server must be installed. The RMA Server files provide the necessary support for RMA on the Content Server.

Follow these steps to install the RMA Server Files:

1. Log in to the **Content Server** host as the **Content Server** installation owner.
2. Copy the RMA server files program, **Records_Manager_Administrator_Content_Server_Files_6.0_x_windows.zip**, to a temporary directory on the Content Server host.
3. Unzip the file.
4. Double-click **rmaServerWinSetup.exe**.
   The **Welcome** dialog box is displayed.
5. Click **Next**.
6. Choose **I Agree** to the license terms.
7. Click Next.
8. Follow the on-screen prompts to finish the installation.
9. Restart the Documentum Java Method Server from the services applet in the Control Panel.

Installing RMA Content Server files on Unix/Linux hosts

Follow these steps to complete the RMA Server Files installation on a Unix or Linux host:

1. Log in to the Content Server host as the Content Server installation owner.
2. Copy the RMA server files program, Records_Manager_Administrator_Content_Server_Files_6.0_x_xx.tar (where .xx is the particular platform specific environment—Solaris, HPUX etc.), to a temporary directory on the Content Server host.
3. Decompress the file. To decompress, type: tar -xvf <filename>
4. Run the rmaServerxxSetup.bin program.
   The files are installed on the Content Server host.
5. Click Next.
6. Choose I Agree to the license terms.
7. Click Next.
8. Follow the on-screen prompts to finish the installation.
9. Disconnect from the Content Server host.

Install docapps

Instructions in this section can be used to install RMA docapps on either Windows or Unix/Linux hosts.

Although you can install up to 9 DocApps for a complete installation of RMA, the first 4 listed below are mandatory to get RMA up and running. **The same docapps must be installed on each repository where RM is installed.** The DocApps must be installed in the order listed:

1. RPS DocApp (rpsdocapp.tar), mandatory
2. PRM DocApp (prmdocapp.tar), mandatory only if PRM is installed
3. RM DocApp (rmdocapp.tar), mandatory
4. Forms Builder DocApp (Forms.zip), mandatory on all repositories where RM is installed including the global repository

5. RM Default DocApp (RM_Defaultdocapp.tar), optional but mandatory if the Chap 2 DocApp is needed/used

6. DoD Chap 2 DocApp (rm_dod5015_2_docapp.tar), optional but mandatory if the Chap 4 DocApp is needed/used

7. DoD Chap 4 DocApp (rm_dod5015_4_docapp.tar), optional

8. MessagingApp.tar (optional, needed if installing the RM Outlook Activator)

9. dcodocapp.tar (optional, needed if installing the RM Outlook Activator)

Install the following docapps on the global repository:

a. Forms Builder DocApp (Forms.zip)

b. RM Forms Adapter DocApp (rm_forms_adapter_docapp.tar), mandatory only on the global repository

Install the RM Default DocApp:

- If you intend to create Formal Records and want an example. It is possible to disable the form template once you have created your own templates.

If you are planning to create Chapter 4 formal records, the Chapter 4 DocApp can not be installed independently of the Chapter 2 DocApp and it must be installed only after the Chapter 2 DocApp is installed. An error will occur if the Ch2 DocApp is not installed.

**Note:** The DocApp files must be extracted from the files listed above to a temporary folder location and run by the Documentum Application Installer (DAI) to install the various DocApps. Use the supported version of DAI to install the DocApps. (5.3.5 is the version to use for the 6.0 release)

The various DocApps must be installed in each repository that is to be used with RMA. This can be done from any host that has the Documentum Application Installer (DAI) and that can communicate with the Content Server.

The instructions below demonstrate the first DocApp installation (the RPS DocApp). Follow the same process to install the remaining mandatory DocApps (or any optional DocApps) in the order specified above. You must install the four mandatory DocApps in each repository that is to be used with RMA.

**To install the RPS DocApp in a repository:**

1. Connect to a Windows host where the supported version of DAI is installed.

   Ensure that the dfc.properties file under C:\Documentum\config points to a DocBroker (Connection Broker on the Content Server) that connects to the
Unix/Linux or Windows repository. You can verify this according to the step used to modify the \texttt{dfc.properties} file described in the \textit{Modify the dfc.properties file, page 75}.

2. Extract \texttt{rpsdocapp.tar} to a temporary folder on a server that has the supported version of DAI installed.

3. Perform the following substeps to install the RPS DocApp.
   a. Click \textit{Start > Programs > Documentum > Application Installer}.
   b. Connect to the repository as the installation owner.
   c. Browse to the temporary folder called \texttt{rps} where the \texttt{rpsdocapp.tar} file was extracted.
      Your selection appears in the Application Installer window.
   d. Click \texttt{OK} to proceed to the installation screen.
   e. Click \textit{Start Installation} from the Application Installer window to begin the process of updating the DocApp for the repository.
      The installation is complete when the \texttt{Quit installer} button becomes available.
   f. View the file \texttt{rps_installerLog.html} for any errors.

4. Repeat the process in order to install the remaining mandatory DocApps and if needed any of the optional DocApps.

\section*{Prepare the application server}

Make sure to prepare the application server, if it has not already been done, according to the applicable instructions in one of the sections under \textit{Chapter 3, Preparing the Application Server Host}.

Update the \texttt{web.xml} file for Tomcat according to the section \textit{Preparing Apache Tomcat} for example.

Make sure to restart the application server.

\section*{Deploy the RMA war file}

Deploy the \texttt{rma.war} file on the applicable application server, such as BEA or Tomcat, to the \texttt{rma} directory. This step depends on the application server chosen by the customer. For Tomcat on Windows, the default location is \texttt{C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\rma}. 
For Tomcat, it is recommended to create a directory for the deployment under webapps such that the name of the directory contains part of the URL, rma for example. Unzip the war file into this directory.

**Modify the dfc.properties file**

Modify the dfc.properties so it is configured with the correct values for the docbroker name and the global registry docbase name.

On Tomcat for example, update the dfc.properties file in C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\rma\WEB-INF\classes and uncomment each line: #dfc.docbroker.host[0]= and set to your docbroker.

The best method to perform this and avoid mistakes, is to copy the 4 lines from the generated dfc.properties file on the Content Server under C:\Documentum\config.

```plaintext
dfc.docbroker.host[0]=<My Server>
dfc.globalregistry.repository=<My Repository>
dfc.globalregistry.username=dm_bof_registry
dfc.globalregistry.password=<My Password>
```

- Each web application has its own copy under WEB-INF\classes.
- Edit the file correctly and specify the docbroker host.
  - Ensure to uncomment the lines (remove #).
  - The global registry repository name is case sensitive.
- The Global registry password needs to be in encrypted form.

Here is an example:

```plaintext
dfc.docbroker.host[0]=mydocbroker
dfc.globalregistry.password=avhSs63+9xY\=
dfc.globalregistry.repository=globalrepo_lab02
dfc.globalregistry.username=dm_bof_registry
```

**Register the DFC client**

Register the dfc client (jar file) from Documentum Administrator (DA) on all RM repositories only (not the global repository) after the application server has been restarted. Login to DA and choose the RM enabled repository. Also, Register the DFC instance on the content server if you need to run any jobs.

You will need to approve the privileged DFC client on each RM repository to complete the registration.
1. Using DA, navigate to Administration > Privileged Clients and select File > Managed Clients.

2. Right-click the applicable client listed according to the Client Name in the content pane and click Approve Privilege.

   For a detailed description of privileged DFC, refer to Content Server Fundamentals.

   For instructions on using Documentum Administrator, refer to the online help or to the Documentum Administrator User Guide.
Chapter 12

Upgrading Records Manager Administrator

Refer to the Records Manager Release Notes, 6.0 before following any upgrade instructions, to verify that your system requirements match or exceed the specifications indicated in the section for the Environment and System Requirements.

The supported upgrade paths for RMA in this release, 6.0 are depicted in the following illustration by the connecting arrows.

Figure 3. Supported upgrade paths

Direct upgrade paths to RMA 6.0 are available for:

• RM 5.2.5 SP5
• RM 5.3 SP4
• RM 5.3 SP5

Any older release must be upgraded incrementally to either RM 5.3 SP4, RM 5.3 SP5, or RM 5.2.5 SP5 before being directly upgraded to RMA 6.0.

Refer to the upgrade procedure listed below to follow instructions for the supported upgrade path:

• Upgrading and Migrating RMA 5.3 SP4 or SP5 to RMA 6.0, page 78
• Upgrading Legacy RM 5.2.5 SP5 to RMA 6.0, page 80

Upgrading and Migrating RMA 5.3 SP4 or SP5 to RMA 6.0

To upgrade RMA on the Content Sever:

1. Backup all 5.3 repositories.
2. Uninstall 5.3 DAB (to view docapps) and DAI (to install docapps) if installed.
3. Do not uninstall the existing Content Server (so as not loose data) unless you wish to wipe out existing data. Upgrade the Content Server according to the Content Server version 6 install guide for the upgrade procedures making certain to perform the following actions as well:
   a. Shut down the Java Method Server and application server if installed.
   b. Shut down the repositories and connection brokers.
   c. Upgrade the Content Server to version 6 according to the Content Server version 6 upgrade instructions.
   d. Upgrade the 5.3 docbroker and all 5.3 repositories.
4. Create a global registry repository, refer to Create global repository, page 70 if necessary.
5. Install DAB version 5.3.5 and DAI version 5.3.5.
6. Install RMA version 6 according to Chapter 11, Installing Records Manager Administrator making certain to perform the following actions as well:
   a. Uninstall or remove RMA 5.3.
   b. Install RMA version 6 according to Chapter 11, Installing Records Manager Administrator.

   Make certain of the following:
   • Turn the pool setting to Off in tomcat\config\web.xml.
• Change the Apache Tomcat properties for the memory to 512M if the web server is tomcat.

c. Delete cookies.

d. Delete the cached files in C:\documentum\cache.

e. Delete all the RMA related jar paths from the Java Classpath in the Apache Tomcat Properties dialog under the Java tab, if the web server is Tomcat.

f. Delete all the RMA related jar paths from the environment variable ClassPath,

g. Delete all the 5.3 RMA jar files in \documentum\shared.

h. Delete the 5.3 RMA jar files in the repository by searching the RMA repository.

To clean up 5.3 unused group and user configurations.

To cleanup RMA 5.3 data on the upgraded RMA 6.0 system:

1. After the RMA 6.0 docapps are installed, copy the RMA_RPS_53_to_60_group_migration.ebs file into the following directory C:\Documentum\product\6.0\bin on the content server.

2. Run the RMA_RPS_53_to_60_group_migration.ebs file in the directory specified above using the following command:

dmbasic -fRMA_RPS_53_to_60_group_migration.ebs
-emigrateRoleConfigFrom53To60 -- <docbase name> <user> <password>.

Replace the <docbase name>, <user>, and <password> fields with the actual values.

To migrate the 5.3 form template:

The new 6.0 docapp installation will cover the form template migration if only the RMA form template originally shipped was used. Users who have their own form template need to open the 5.3 form template in Form Builder version 6 and save it. The template saved will be automatically migrated to the version 6 template. Most of the migration will be done by this procedure though form adaptors need re-configuration. Beware that customizations will be lost for those users using a customized RMA template once the RMA 6.0 template is installed.
Upgrading Legacy RM 5.2.5 SP5 to RMA 6.0

A fresh installation is required to migrate legacy RM data to RMA 6.0. The existing system needs a fresh RMA 6.0 installation to be the target for the data migration. Refer to instructions contained in the Documentum Records Manager Migration Guide, version 6.
Chapter 13

Configuring Records Manager Administrator

Make sure to turn off folder security from Documentum Administrator (DA) when you configure RM.

Note: You need to be in one of the following three roles to create records:

• Records Manager Role, dmc_rm_recordsmanager
• Records Privileged User Role, dmc_rm_privilegeduser
• Records Contributor Role, dmc_rm_recordscontributor

Additionally, you need to be in one of the following two forms roles:

• form_user
• form_designer
Pre-Installation Checklist

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

**Table 7. Preinstallation 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 14.</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>Requirement</td>
<td>For More Information</td>
<td>Completed?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</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>
<tr>
<td>Determine which repositories will be used to store presets and user preferences.</td>
<td>Network administrators</td>
<td></td>
</tr>
<tr>
<td>Determine whether language packs will be required.</td>
<td><em>Web Development Kit Applications Language Pack Installation and Release Notes</em></td>
<td></td>
</tr>
<tr>
<td>Prepare the application server host and application server software according to the vendor’s requirements.</td>
<td><em>Specific requirements are described in Chapter 3, Preparing the Application Server Host.</em></td>
<td></td>
</tr>
</tbody>
</table>
Installing Brava! Webtop Server for RMA

Records Manager Administrator uses the Brava! Viewer to allow you to view the content of a document.

Note: The Brava! Viewer supplied with Records Manager is intended for viewing only. You will not be able to edit, add markups, nor annotate the document being viewed.

The Viewer allows you to view a rendition of a document regardless of whether it is a bitmap image (.BMP), a Microsoft Word document (.DOC), or just about any format of document. That is, when you want to view a document, and you do not have the associated software installed for the document format, RMA opens the document in the Viewer to allow you to view the content of the document.

For detailed step-by-step instructions for installing and configuring the Brava! Server and Client components, see the Brava Enterprise Installation and Configuration Guide, Version 5.3 SP3.

It is important that the Brava Server and Net-It Enterprise are installed on a separate machine from the RMA Server since each has different system requirements.

Note: When installing the Brava! Webtop Server component, the Webtop alias is the context path where you installed RMA. For example, if the URL to RMA is http://<your server name>/rma, then the application server’s context path for RMA is rma. Make sure you type rma, not webtop as shown below.

Figure 4. RMA Webtop alias

Please enter the Webtop alias.

Alias: rma
You can skip most of section 3.0, "Configuring your Installation” of the *Brava Enterprise Installation and Configuration Guide* as it does not apply to RMA. This is because the Brava Server component is already integrated into RMA. However, you MUST run section 3.2, "Installing Brava DocApp” of the *Brava Enterprise Installation and Configuration Guide* to install the custom formats required by Brava to your docbase.

**To complete the Brava! configuration for RM:**

1. Modify the *brava_parameters.jsp* file on the application server where RMA is installed.
   
   For a base Tomcat installation, the file is located in the `C:\Program Files\Apache Software Foundation\Tomcat 5.0\webapps\rma\rm\library\brava` directory.
   
   String serverhostname = "http://<server_name>:<port_number>";
   
   String webAppName = "rma";
   
   Replace the "<server_name>:<port_number>" with the application server name and port number for the RM application server.
   
   Replace the "webAppName" entry with the virtual directory that was used when installing the RMA application. The default is "rma".

2. Replace the license file *IGCKey.lic* with the one provided.
   
   This file is stored in the two following default locations: `C:\Program Files\IGC\Brava! 5.3 Webtop\JobProcessor\LicenseKey` and `C:\Program Files\IGC\Brava! 5.3 Webtop`.
   
   Replace the file with the updated one provided by Documentum and re-start the Net-It Enterprise Server Service and the Application Server service or console.

3. Verify, on Net-It Enterprise Server, that the Black Ice EMF TS printer driver is set as the default printer. To do this, navigate to **Start > Printers and Faxes**. Right click on the Black Ice EMF TS printer and select the **Set as Default Printer** option.
Index

A
Apache Tomcat
  Java heap size, 24
application
tools, 67
Application Connectors
  architecture, 51
  enabling, 52
application server host requirements
  Java heap size, 24
application servers
  performance tuning, 24
  starting, 43
  startup files, 34
  verifying, 43

B
backing up customizations, 33
BEA WebLogic
domains, 25
  Java heap size, 24
  session affinity support, 31
  upgrading, 41
browsers
  Citrix client, 20
  slow display, debugging, 63

C
CA
  configuration, 44
caching
  JavaScript issues, 20
Citrix client, 20
ClearTrust
  configuration, 44
clients
  preparing, 17
  set JVM, 17
customizing applications, 9
  backing up customizations, 33
developer licenses, 15
clustered environments, Oracle
  Application Server, 29
collection stripper, 67
collection, typical, 12
collection troubleshooting, 62
collection broker
troubleshooting, 61
connection brokers, 38
  deployment requirement, 11
Content Server
  deployment requirement, 11
  requirements, 13
  versions, 13
Content Server requirements
global registry, 13
content transfer
  enable in IE7, 18
  enable in Firefox, 19
temporary directory, 14
content transfer operations
Documentum Application
  Connectors, 51
CreateInstallerWAR tool, 68
customizing an application, 15

D
default web applications, 47
deploying
  application server host
    requirements, 14
customizing an application, 15
  multiple applications, 15
  planning, 11
  required directories, 14
  single application server, 12
  supporting software, 11
typical configuration, 12
Web Development Kit application, 37
deployment
  completing the process, 43
testing, 49
developer licenses, 15
developing applications, 15
DFC
global registry, 38
dfc.properties, 39
  connection broker, 38
directories
  content transfer, 14
  permissions, 14
DNS
  requirement, 15
DocApps, 11
  requirement, 14
docbroker
  troubleshooting, 61
Documentum Administrator
  customizing, 15
Documentum Application Connectors
  command-line installation, 54
  content transfer operations, 51
  GUI installation, 52
  uninstalling, 55
domains, BEA WebLogic, 25

E
environment
  variables, 24
extended search DocApps
  installing, 57
external web servers, 31

F
Firefox
  preparing for content transfer, 19
forward proxy
  preparation, 31

G
global registry, 38
  requirement, 13
  troubleshooting, 61
global security on IBM WebSphere, 44

I
IBM WebSphere
  global security, 44
  Java heap size, 24
  predeployment requirements, 26
  session affinity support, 31
installation owner
  changing, 36
  content transfer directory, 14
  required permissions, 14
  WDK application, 36
installing
  application server software, 23
  DocApps, 11
  extended search DocApps, 57
  host requirements, 12
  virtual link support, 47
Internet Explorer
  Windows XP SP2, 21
  Internet Explorer 7
    preparing for content transfer, 18

J
Java
  heap size, 24
  memory allocation values, 24
Java heap
  MaxPermSize parameter, 24
JavaScript
  clearing cache, 20

L
language packs, 15
localization, 15
login page
  troubleshooting, 62

M
MaxPermSize parameter on BEA
  WebLogic, 24
memory
  dfc.properties, 39
multiple applications, deploying, 15

O
Oracle Application Server
  clustered environment, 29
Java heap size, 24
predeployment requirements, 29
WebCache, 29
Oracle WebCache, 29
out of memory errors, 24

P
performance
  DFC setting, 39
tuning, 24
planning for deployment, 11
policies
  Sun Java System Application Server, 30
pop-up blockers, 18
predeployment requirements
  IBM WebSphere, 26
  Java heap size, 24
  Oracle Application Server, 29
  Sun Java System Application Server, 30
  Tomcat, 25
  WebLogic domain, 25
preferences
  repository, 40
preinstallation requirements
  application server software, preparing, 23
preparing
  application server host, 23
  client JVM, 17
  clients, 17
presets
  repository, 40
proxy server
  preparation, 31

R
repository
  for presets and preferences, 40
required directories
  content transfer, 14
reverse proxy
  preparation, 31
root user, 36
RSA
  configuration, 44

S
session affinity support, 31
single sign-on
  configuration, 44
SiteMinder
  configuration, 44
SSO
  configuration, 44
startup files, application server, 34
Sun Java
  plugin, 17
Sun Java System Application Server
  predeployment requirements, 30

T
tag pooling
  Sun Java System Application Server, 30
troubleshooting, 64
Tomcat
  predeployment, 25
tools
  deployment, 67
trusted Sites, 18
typical configuration, 12

U
UCF content transfer, 17
uninstalling
  application installed by another user, 36
  DFC, 35
Documentum Application Connectors, 55
  older software versions, 34
  overview, 34
  WDK-based application, 35
upgrading
  application server startup files, 34
  backing up customizations, 33
  BEA WebLogic, 41
  overview, 33
  root user, 36
  WDK applications, 34

V
variables
environment, 24
viewing WDK samples, 49
virtual link support
  in 5.3 and later installations, 47
  legacy support, 47

W
WAR file
  preparing for deployment, 37
WAR files
packaging tool, 68
WDK applications
  accessing, 48
  deploying, 37
  verifying, 48
web servers, external, 31
Webtop Extended Search DocApps
  log files, 60
Windows
  XP SP2, 21