EMC® Documentum®
Documentum Administrator

Version 6.5 SP1

Deployment Guide

P/N 300-008-384 A01

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 017489103
1-508-435-1000
www.EMC.com

Copyright© 1994 - 2008 EMC Corporation. All rights reserved.

Published December 2008

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

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

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

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Table of Contents

List of Figures

List of Tables

Preface

This guide describes how to deploy the Documentum Administrator application.

Intended Audience

This guide is intended primarily for administrators who are deploying Documentum Administrator.

You should be familiar with the Windows, UNIX, or Linux operating system and be able to install and configure a J2EE application server.

Documentum Administrator is a WDK-based application. The deployment process is largely the same as for other WDK-based applications. The steps specific to deploying Documentum Administrator begin in Chapter 10, Deploying Documentum Administrator.

Revision History

The following changes have been made to this document:

Revision History

Revision Date	Description
December 2008	Initial publication

Chapter 1

Quick Start

This chapter describes the steps you need to perform to deploy your application. The steps are described in more detail in the chapters of this guide. Your product or environment may require additional steps, which you can find in the product-specific chapter or chapters of this guide or in the index.

To perform a simple product deployment

1. Plan the deployment. (Refer to Chapter 2, Planning for Deployment.)

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

2. Prepare the clients. (Refer to Chapter 4, Preparing the Client Hosts.)

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

3. Prepare the application server. (Refer to Chapter 5, Preparing the Application Server Host.)

   Ensure you have sufficient memory allocated to the application server Java instance, turn off failover if it is not needed, and follow application-server and proxy-server specific preparation instructions.

4. Deploy the product WAR file using the application server standard deployment mechanism. (Refer to Chapter 7, Deploying a WDK-based Application.)

   You must first unpack the WAR file archive and enter some information that is specific to your environment: your connection broker and global registry information, optional presets and preferences repositories, and optional Federated Search Server.

5. Complete the deployment. (Refer to Chapter 8, Completing the Deployment.)

   After successful deployment, you can configure UCF, deploy root virtual link support, enable WebSphere global security if needed, and test the application samples.

6. Complete the steps specific to deploying Documentum Administrator. (Refer to Chapter 10, Deploying Documentum Administrator.)

Chapter 2

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

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

- Content Server and its associated database
- Global registry
- Connection broker

  You must specify one or more connection brokers in the dfc.properties file. Refer to To configure connections in dfc.properties before deployment, page 36 for information on configuring the docbroker before deployment.
- J2EE application server or servlet container

The Webtop DocApps are provided in Content Server version 6 or higher.

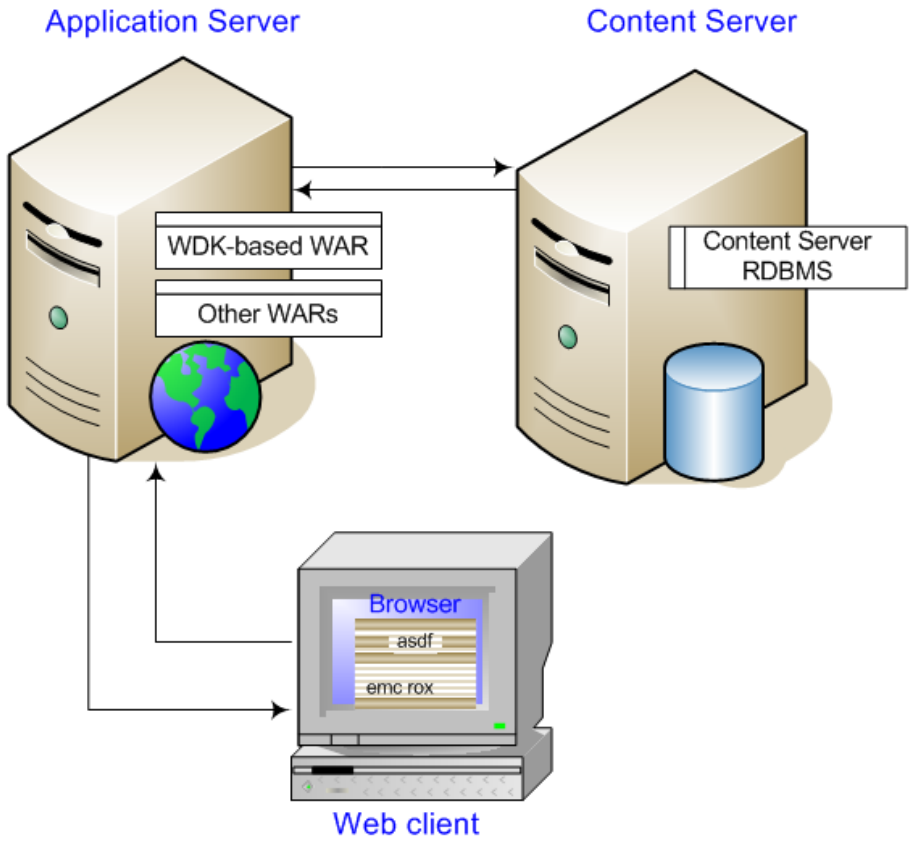
Typical configuration

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
- Client hosts that run a supported web browser

Figure 1, page 14 shows the network components.

Figure 1. Basic WDK host configuration



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

Clustered environments — WDK-based applications can be deployed in supported clustered environments. Refer to the release notes for each WDK-based application to learn which managed server configurations are supported.

Preparing the Content Server

The following topics describe Content Server requirements.

Content Server 6 or higher installs certain DocApps that are required for a WDK-based application. Webtop and DA DocApps are installed by default.

The global registry requirement — A global registry of Content Server version 6 or higher 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 36.

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 of version 6 or higher 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.

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 3

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

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

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

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

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

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

Supporting WDK 5.3.x and 6 or higher applications on the application server — A 5.3.x application cannot share the same instance as a version 6 or higher application. One or both of the applications will not work properly.

Chapter 4

Preparing the Client Hosts

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

This chapter contains information on the following predeployment tasks:

Ensuring a certified JVM on browser clients

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

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

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

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

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**.
3. Scroll to the **Downloads** section and enable **Automatic prompting for file downloads**.
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

Firefox 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 JavaSript**, and **Enable Java**.
6. Install the Firefox add-on FireBug, which is available from mozilla.org.

Configuring Firefox version 3.x

In the Firefox version 3.x, the file browse tag <input type='file'…./>, by default, returns the file name only (not the absolute path). See the link for more information https://developer.mozilla.org/en/Updating_web_applications_for_Firefox_3#File_upload_fields, whereas the Documentum Administrator application require the absolute file path.

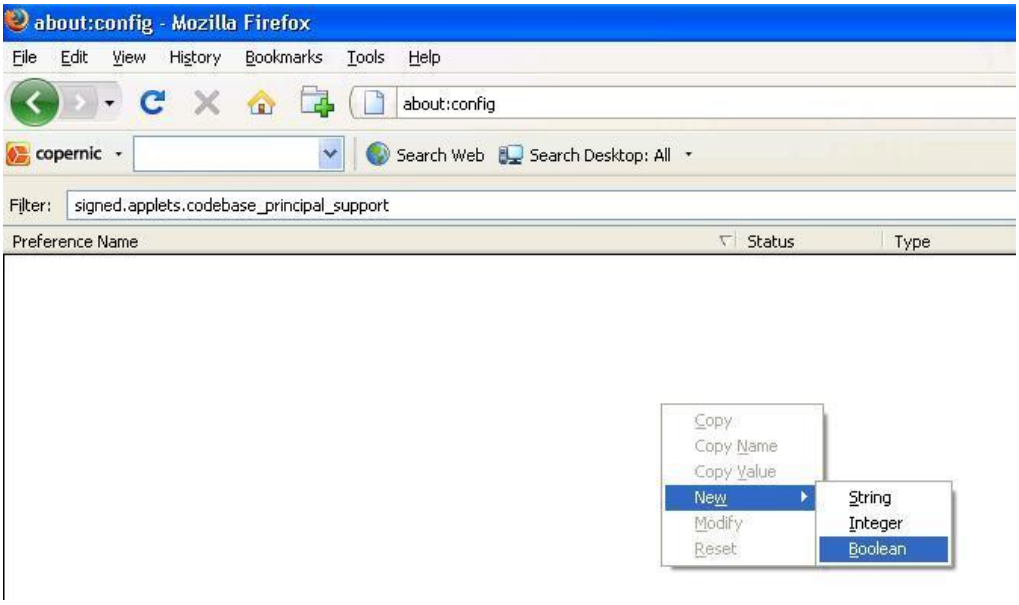
To configure Firefox version 3.x browser:

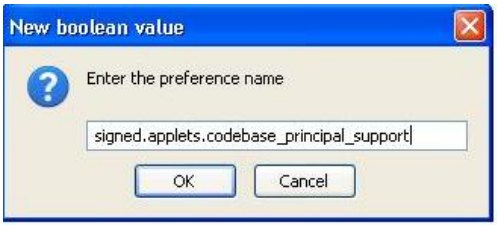
1. Add a new preference called 'signed.applets.codebase_principal_support' (without quotes) and set its value to **true**. This configuration can be done by the end user or IT can push the configuration to all the end users.

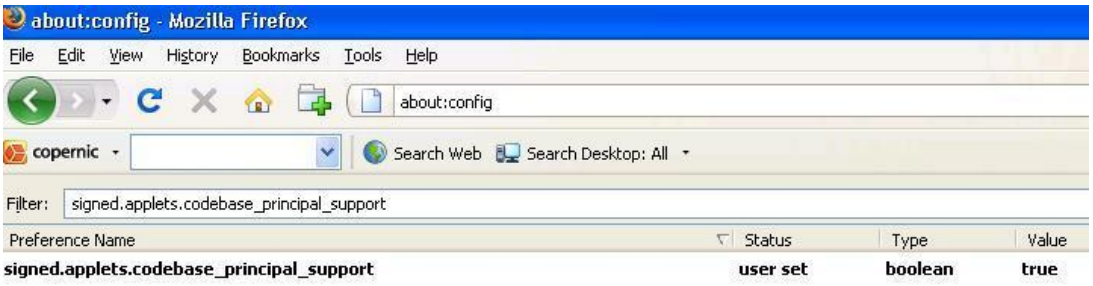
   a. For end user configuration:

   - Type about:config in the browser location bar, press enter and accept the warning message as shown in the picture.

- Search for 'signed.applets.codebase_principal_support', if the preference is found, and make sure its value is set to true. If the preference is not available, add a new preference as shown in the pictures and set its value to **true**, close the browser and relaunch. Verify whether the preference is set to true.







b. For IT Push procedure, in the Customer site, the IT team can push the configuration to all the Firefox/Documentum Administrator users, to avoid individual users to set this. Follow the steps below for IT push:

- Create a file named user.js and add the following details in the file.

  user_pref("signed.applets.codebase_principal_support",true);

- IT can push user.js file into the user profile location of Mozilla\Firefox. For more information on IT push/preference, an update is available at https://developer.mozilla.org/En/A_Brief_Guide_to_Mozilla_Preferences

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

  The user profile directory location for Mozilla\Firefox is different in various operating systems. For more information, see http://kb.mozillazine.org/Profile_folder.

2. For Documentum Administrator, the end user must allow/accept when the Internet Security alert dialog gets popped up while using the File Browse control on Documentum Administrator UI.

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 ES1_MRE.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 ES1_MRE.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 and Windows XP SP3 installs a pop-up blocker in Internet Explorer that is enabled by default. HTTP content transfer operations in WDK applications are prevented by the pop-up blocker. You must turn off the pop-up blocker for HTTP transfer.

Chapter 5

Preparing the Application Server Host

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

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

Before you deploy a WDK-based application, ensure that your J2EE application server or servlet container is a supported version and that it can successfully serve sample JSP pages.

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

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

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

Setting the Java memory allocation

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

```
-Xms1024m -Xmx1024m
```

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

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

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

```
-server
-XX:+UseParallelOldGC
```

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

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

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

Turning off failover

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

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

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

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

Preparing environment variables for non-default DFC locations

The base location for content transfer on the application server host is specified by the DFC environment variable dfc.data.dir. This location is specified as the value of the key dfc.data.dir in

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 and JBoss server

Please refer to your product release notes to determine whether Apache Tomcat and JBoss are supported application servers 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:

```
<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>
  <init-param>
    <param-name>fork</param-name>
    <param-value>false</param-value>
  </init-param>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>false</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>
```

Note: The location of the web.xml for JBoss is jboss install location/server/default/deploy/jboss-web.deployer/conf/web.xml

Preparing WebLogic Server

Please refer to your product release notes to determine whether WebLogic servers are supported application servers for your product.

The following topic describes preparations that may be necessary before you deploy a WDK-based application.

Supporting large content transfer operations in a managed server environment

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

Preparing IBM WebSphere

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

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

Supporting failover in a cluster

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

Applying policies for WebSphere security

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

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

- dfc.data.dir

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

- webtop.content.xfer

  Specifies the temporary content transfer directory on the application server. Must match the value in app.xml of the element <contentxfer>.<server>.<contentlocationwindows> or <contentlocationunix>.

The policy files in PolicyFiles.zip contain the minimum required policies for the dfc.data.dir directory. To add additional policies for non-default content transfer locations, add the following lines to library.policy. For each policy that you add, set up an environment variable that specifies the non-default location.

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

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

```
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.

```
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.

```
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) —

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

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

```
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";
```

Preparing Oracle Application Server

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

You must disable tag reuse for the application server.

To disable tag pooling for the application

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

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
  <init-param>
    <param-name>tags_reuse_default</param-name>
    <param-value>none</param-value>
  </init-param>
</servlet>
```

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:

```
<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

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServl
  <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>
```

Turning off failover

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

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

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:

```
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 6

Upgrading 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 a WDK application. Customization of Documentum Administrator is not supported.

Table 1, page 33 shows the files, directories, and subdirectories on the application server host that should be backed up.

Table 1. Directories and files to back up

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

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*.

Chapter 7

Deploying a WDK-based Application

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

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

The following topics describe how to deploy the application:

Preparing the WAR file for deployment

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 36. 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 37.
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 or higher is required for WDK-based applications. The global registry is a central repository that serves several purposes:

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

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

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

To find the essential dfc.properties file values from the Content Server installation

1. On the global registry 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
```

To configure connections in dfc.properties before deployment

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

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

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

   ```
   dfc.docbroker.port=port_number
   ```

5. Add the global registry name to the following key:

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

6. Add the username of the dm_bof_registry user to the following key:

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

   The global registry user, who has the username of dm_bof_registry, has read access to objects in the /System/Modules and /System/NetworkLocations only.

7. Add an encrypted password value for the following key:

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

   You can either copy the username and encrypted password from the dfc.properties file on the global registry Content Server host, or you can select another global registry user and encrypt the password using the following command from a command prompt (assumes the directory containing javaw.exe is on the system path):

   ```
   java -cp dfc.jar com.documentum.fc.tools.RegistryPasswordUtils
   password_to_be_encrypted
   ```

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.

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 36 for the procedure of unpacking the war file and modifying dfc.properties. Add the following line to your dfc.properties file:

```
dfc.diagnostics.resources.enable=false
```

Enabling presets and preferences repositories

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

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

Table 2. Preferences configuration elements

Element	Description
<preferencesrepository>	Contains a <repository> element. If this element is not present, user preferences are stored in the global registry, which can slow down performance.

Element	Description
.<repository_path>	Specifies the path within the preference repository in which to store preferences. If the path does not exist at application startup, it will be created.
.<repository>	Specifies the repository in which to store preferences, preferably not the global registry.

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 Federated Search Server. (The Federated Search Server is a separate product that is purchased separately from Webtop and Content Server.) If this connection has not been configured, you cannot include external sources in your search.

To configure the connection to Federated Search Server

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

Deploying multiple applications

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

Chapter 8

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

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

- How to configure different content transfer mechanisms (UCF or HTTP) for roles.

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

  Note: The web server associated with an application server must support chunked requests. The web server forwards HTTP requests using chunked transfer encoding, as described in the 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.

Configuring IBM WebSphere after deployment

To complete the deployment, perform the following procedures.

Manual steps for deploying Documentum Administrator on IBM WebSphere

The following manual steps need to be performed after deployment:

1. Set '**com.ibm.ws.webcontainer.invokefilterscompatibility**' custom property to '**true**' under Application Servers > Server1 > Web container > Custom Properties in Admin console.
2. Add 'dfc.diagnostics.resources.enable=false' additional parameter in dfc.properties file of Documentum Administrator application.
3. Copy the xml.jar from Websphere/AppServer/java/jre/lib directory to Documentum Administrator application WEB-INF/lib directory.

   Changing the classloader and compiler settings:
4. 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**.
5. 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/da_war.ear/deployments/da/da_war/
   da.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"/>
   ```
6. Restart the application server.

Configuring Oracle Application Server

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

```
<init-param>
     <param-name>wdk_cache_control_redirect_includepages</param-name>
     <!-- ending with ($) certain file extensions -->
  <param-value><![CDATA[(\.jar)$]]></param-value>
</init-param>
```

Deploying default virtual link support

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

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

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

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

To install default virtual link support

1. Deploy the vlink.war file as the root web application on the application server.

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

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

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

   On Weblogic, add the following line to weblogic.xml file or use the weblogic.xml file that is bundled with vlink.war:

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

For more information on virtual links, refer to the *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

   `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 Content Server through the WDK-based application. Content Server provides the connection to the repository.

   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 Content Server. 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 9

Configuring Single Sign-On for Security Servers

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

Content Server supports pluggable authentication or SSO using RSA Access Manager (formerly known as ClearTrust) or CA SiteMinder.

RSA Access Manager users must have the same login names as the Content Server. User names are case-sensitive for the Content Server, so Access Manager user names must be at least 8 characters in length and have the same case as the Content Server 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.

To enable single sign-on (SSO):

1. Configure the RSA Access Manager 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 46.
5. RSA only: Create a directory named rsaConfig under the root WDK-based application directory. Copy two files: aserver.conf from the Access Manager server and webagent.conf from the RSA web agent. Paste them into the rsaConfig directory.

   If you make changes to the original files, you must copy the changed files to your WDK-based application rsaConfig directory. For more information on these files, refer to the RSA documentation.
6. Locate the file AuthenticationScheme.properties in WEB-INF/classes/com/documentum/web/formext/session. The SSO authentication scheme classes. Modify the properties file to make your preferred SSO authentication scheme (SSOAuthenticationScheme or 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 is presented with a login screen instead of single sign-on.

7. Restart the application server.

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

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 to be retrieved from a vendor-specific cookie.
- Username, which is retrieved from a vendor-specific HTTP requests header or remote user.

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

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

Note: This example is for RSA.

Table 3, page 47 describes valid values for each element.

Table 3. Authentication elements (<authentication>)

Element	Description
<docbase>	Specifies default repository name. When SSO authentication is enabled but a repository name is not explicitly spelled out by the user nor defined in this element, the sso_login component is called. In this case the component prompts the user for the repository name.
<domain>	Specifies Windows network domain name.
<service_class>	Specifies fully qualified name of class that provides authentication service. This class can perform pre- or post-processing of authentication.
<sso_config>	Contains SSO authentication configuration elements.
<sso_config> <ecs_plug_in>	Specifies name of the Content Server authentication plugin (not the authentication scheme name). Valid values: RSA: dm_rsa CA: dm_netegrity
<sso_config> <ticket_cookie>	Specifies name of vendor-specific cookie that holds the sign-on ticket. Valid values: RSA: CTSESSION CA: SMSESSION
<sso_config> <user_header>	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 LegacyVariable flag is set to true or false. If false, use SMUSER, if true, use SM_USER.

Chapter 10

Deploying Documentum Administrator

The following sections describe requirements for deployment of the Documentum Administrator application.

About Documentum Administrator

Documentum Administrator is a Content Server and repository administration tool. Use Documentum Administrator to create users, groups, permission sets, administrator access sets, federations, configuration objects (ACS, BOCS, LDAP, and server, for example), site publishing configurations, types, formats, storage areas, and alias sets. You can also use Documentum Administrator to stop and start servers, run jobs, methods, and administration methods, create new jobs and methods, and administer full-text indexing.

Documentum Administrator includes all Webtop content-management functionality. In addition, other Documentum applications are administered using Documentum Administrator, including Site Caching Services, Retention Policy Services, Resource Management, and Content Transformation Services.

Customizing Documentum Administrator is not supported. Appserver Cluster for clustered environments is not supported in Documentum Administrator.

Note: The following functionality is present only in Content Server version 6 or higher and cannot be provided in a DocApp:

- Administrator access
- Privileged clients
- LDAP failover
- Distributed content
- Resource management

Fully-qualified domain name required for Documentum Administrator

If you use Documentum Administrator to administer full-text indexing, the host where the application server is installed must be identified by a fully-qualified domain name. For example, the host name tristan.documentum.com is acceptable, but an IP address (for example, 123.45.6.789) is not acceptable.

Resource Management availability

If Resource Management is installed, the RMI port used to manage the resources must be open. If a firewall separates the machine hosting Documentum Administrator from the remote resource, the RMI port must be open and not obstructed by the firewall. Also, the Domain Name Server must be configured to properly resolve IP addresses based on the URL used to access the server.

Enable presets for Administrator Access and Resource Management

When deploying Documentum Administrator, the **Enable/Disable Presets** flag in the application custom app.xml file must be set to True, as it impacts the following functionality:

- **Administrator Access**: If the preset flag is disabled, the Administrator Access functionality in Documentum Administrator is disabled.
- **Resource Management**: If the preset flag is disabled, the ability to dynamically access or modify the resource agent information in the global registry is disabled. Resource Management will still function for resource agents defined in the static configuration file, but administrators will not be able to add, modify, or delete resource agents using Documentum Administrator.

The Enable/Disable Presets flag in the custom app.xml file for Documentum Administrator overrides the presets flag in WDK.

Manual step for configuring LDAP SSL

The LDAP SSL functionality requires a manual step in the Documentum Administrator installation process. To complete the manual step to configure LDAP SSL in Documentum Administrator, refer to the documentation in the application deployment directory under WEB-INF\thirdparty\readme.txt.

Modal popup

Modal popup is supported only on browser environment, where only Internet Explorer browser is supported. It is not supported in the 508 accessibility mode.

Overview

When you invoke a component that has been configured for display in modal popup through action definition or others, the User Interface for the component is displayed in a modal popup window. This modal popup window is placed on top of the current window. It is positioned on the center horizontally relative to the parent window and vertically relative to the screen. The title of the modal popup window shows the title of the component page followed by "— Webpage Dialog". You can resize the modal popup window, but won't be able to access the parent window until you dismiss the popup window (also known as child window). When you try to close a modal popup window by clicking the [X] button on the window, the framework treats it as a cancel.

In addition, when you invoke another component that is configured for display in modal popup from the child window, another modal popup window is placed on top of the child window to show the component's User Interface. At this point, you will see stacked modal windows and you cannot access a parent window until you dismiss its child window(s).

Configuring the modal popup

You, as a developer, can configure whether a nested component is to be displayed in a modal popup. If a component is tied to an action, then you can modify the action definition, by adding the <invocation> element.

```
<action id="about">
 <params>
  <param name="enableTools" alias="CtrlKeyPressed" required="false"
 </params>
 <execution class="com.documentum.web.formext.action.LaunchComponent">
   <component>about</component>
 </execution>
 <invocation>
  <modalpopup>
   <windowsize>small</windowsize>
   <refreshparentwindow>never</refreshparentwindow>
  </modalpopup>
 </invocation>
</action>
```

This configuration is added to the action definition because the modal popup behavior is more tied to how a component is invoked. Here, the idea is to have the modal popup configuration in the action definition. In the invocation element, you can specify the size of the modal popup and whether the framework should refresh the parent window when the child window is closed. All action controls will read the configuration. During action invocation, if the configuration indicates that the component tied to this action should be displayed in a modal popup, this will open a modal popup window and submit the request to the component. The response is then displayed in the modal popup window.

Chapter 11

Troubleshooting Deployment

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

Wrong JRE used for application server

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

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

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

No global registry or connection broker

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

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

To fix this error, either provide the correct BOF registry connection information in dfc.properties, or do not provide any connection information at all. Refer to the *Content Server Installation Guide* for information on enabling a repository as a global registry.

No connection to Content Server

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_FAIL
ED] 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 Content Servers. Refer to To configure connections in dfc.properties before deployment, page 36 for information on enabling the connection in dfc.properties.

If the Content Server that is specified as the global registry is down, the following message may be displayed:

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

DM_VEL_INSTANTIATION_ERROR

This error can be caused by several setup problems:

- Not using a version 6 or higher global registry
- Installing DAB 5.3 on the same machine as the application server

Login page incorrectly displayed

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

If the login page displays several controls with the same label, you have not turned off tag pooling in the application server. Refer to Tag pooling problem, page 55 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 26.

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

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

Slow display first time

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

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

DFC using the wrong directories on the application server

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

Application startup errors

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

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

Tag pooling problem

If you have not properly disabled tag pooling in the application server, you will see several instances of the same control on the login page. For instructions on disabling pooling in Tomcat, refer to Preparing Apache Tomcat and JBoss server, page 27. For the Sun Java System server, refer to Turning off tag pooling, page 31. For Oracle, refer to Preparing Oracle Application Server, page 30.

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

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

UCF client problems

If the error message "Compatible Java Run time environment is not installed" is displayed on a non-Windows client, make sure that you have installed version 1.5.0_06 of the Sun JRE on the client; this version will be used by UCF and will not interfere with the browser VM. The client browser VM must be one that is certified in the release notes. It will be used for non-UCF applets.

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

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

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:

```
<defaults>
  <ucfHome value="$env(USERPROFILE)/Documentum/ucf"/>
  <ucfInstallsHome="$env(USERPROFILE)/Documentum/ucf"/>
  <configuration name="com.documentum.ucf">
     <option name="user.dir">
       <value>$env{USERPROFILE}/Documentum</value></option>
```

3. Save and restart the application server.

Connection issues between an Federated Search Server and IPv6 clients

Federated Search Server uses the RMI protocol to communicate with the client applications. When the client application launches a request against the Federated Search Server, it indicates the IP address that the Federated Search Server should use to respond. However it may happen that the client sends a link-local address instead of a global address. To avoid any connection issue, you can update the catalina.bat script that launches the WDK application. The following setting forces the RMI IP to connect:

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

Appendix A

Pre-Installation Checklist

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

Table 4. Preinstallation tasks

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

Requirement	For More Information	Completed?
Determine which repositories will be used to store presets and user preferences.	Network administrators	
Determine whether language packs will be required.	*Web Development Kit Applications Language Pack Installation and Release Notes*	
Prepare the application server host and application server software according to the vendor's requirements.	Specific requirements are described in Chapter 5, Preparing the Application Server Host.	

Index

E

F

G

I

J

L

M

O

P

R

S

T

U

V

W