EMC® Documentum® Foundation Classes

Version 6.7

Installation Guide

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

Table of Contents

List of Tables

Preface

This manual describes how to install EMC Documentum Foundation Classes (DFC).

Intended audience

This manual is for programmers or system administrators who must install DFC. For Windows systems, it assumes general familiarity with Windows operation. For UNIX systems it assumes general familiarity with shells, permissions, and environment variables.

Document Scope

This document addresses the rare case where you would install DFC as a standalone product. This would be useful primarily to customers who have created their own interface to the Documentum system. Documentum applications each install their own instance of DFC. For information on updating and maintaining DFC with client applications, see the deployment guide for the particular application you're using.

Revision History

The following changes have been made to this document.

Revision History

Revision Date	Description
April 2011	Initial publication

Chapter 1

Before You Install DFC

This chapter describes the steps you must take before installing DFC. Any EMC Documentum client product that uses DFC installs DFC. However, you can upgrade DFC without upgrading the client program that installed DFC. You might do so to incorporate the bug fixes or other improvements that we provide in a service pack.

Caution: If you previously installed additional classes on top of DFC under the DFC program root directory, check whether those classes are still in place after you upgrade DFC. For example, some InputAccel use cases required that you install bpmutil.jar and bpm_infra.jar after you installed DFC. Another example, you may have customized your system and created custom classes and installed them in the DFC shared directory after installing DFC.

Whenever you upgrade DFC, you will need to check if your additional classes need to be reinstalled after the upgrade.

DFC relies upon certain environment variables. Whether you install DFC directly or rely on a client program to do so, you must ensure that these variables have correct values. Establishing the environment for DFC, page 9 explains how to set the environment variables that DFC relies upon.

DFC requires administrator privileges to run. At installation, the installer verifies that the installation account has those privileges.

This chapter contains the following sections:

Where to install DFC

You can install DFC on:

- A middle-tier system.

For example, to support WDK or Content Server methods.

- An end user's computer.

  For example, to support Desktop.

DFC runs on a Java virtual machine (JVM) on the machine from which you call it.

Note: Refer to the DFC release notes for the supported versions of the JVM. These can change from one minor release to the next. Using DFC with an application server may further restrict the supported versions.

Java 2 Security

If you plan to use the Java 2 security architecture for your installation, you will need to create security policy files to specify the permissions allowed to the classes in your installation. To help you develop your own security policy files, an example policy template file is included in the DFC installation. The file specifies the permissions required for the DFC classes. You will need to merge these permissions to your policy file, and replace the variables in the file with the values in your installation to allow the DFC classes to function. You will specify the policy file that you create at the startup of your Java Virtual Machine (JVM). For a stand-alone installation, you will specify the file in the command line startup arguments for your JVM. If you are deploying on an application server, refer to the application server documentation for the specifics of specifying Java 2 security policies. The DFC example policy template file is named dfc.example.java.policy, and is found in the dfc.jar file installed in the DOCUMENTUM_SHARED directory (by default, in Windows systems, C:\Program Files\Documentum\Shared).

Whether to remove old programs first

This section explains the situations in which you must remove an older version of DFC, and possibly other programs as well, before installing this version of DFC. It contains the following sections:

- Whether to remove the old DFC, page 8
- Whether to upgrade client programs, page 9

Whether to remove the old DFC

You can install this release of DFC directly over DFC version 5.3 or later.

Install the new DFC without uninstalling the old one if the following conditions are true:

- The DFC installation you want to upgrade has a version number of 5.1 or later.
- The DFC installation is on a Content Server host.

In all other cases, uninstall the old DFC before installing the new one.

Uninstalling DFC, page 14 explains how to remove DFC.

Whether to upgrade client programs

Programs that use DFC are called client programs. Upgrading DFC does not affect client programs that use DFC version 5.3 or later.

In order to run more than one version of DFC on a UNIX system, you must arrange to run the different DFC versions in different processes. Set the environment variables described in Establishing the environment for DFC, page 9 and install the different versions of DFC in locations that you can distinguish from one another by those environment settings.

Establishing the environment for DFC

This section explains how to set the environment variables that DFC relies upon.

Table 1, page 11 lists the environment variables that DFC relies upon. To set the variables:

- **Windows:** Installation program sets the environment variables.

  The DFC installation program for Windows sets environment variables. The only additional setting you need to make is to add jars to the classpath if you need to refer to DFC classes and interfaces in your Java programs. Locations of DFC classes, page 11 provides more details about what to place on the classpath.

- **UNIX:** You set environment variables.

  For UNIX systems, the installation program does not set environment variables. If the installation program does not find the needed environment variables, it aborts the installation.

  For UNIX systems, the way to set environment variables depends on the shell that you use. Be sure to set the variables in such a way that a process launched in a different shell has the same values defined. This means using setenv or export (depending on the shell). Do not use *set*, which defines variables only for the current shell, but not for any child shell.

The following topics describe the considerations in setting up the DFC environment for Windows or UNIX:

- Defining file system locations for DFC components, page 9
- Setting environment variables, page 11
- AIX only: Correcting the jaxp.properties file, page 13

Defining file system locations for DFC components

DFC maintains components at different file system locations. The following sections provide details about the locations DFC uses.

DFC program root directory

DFC installs program files under the program root directory.

On Windows systems, the installation program asks for a program root directory. It uses C:\Program Files\Documentum if you do not specify a location.

On UNIX systems, the installation program uses the environment variable DOCUMENTUM_SHARED to determine the program root directory. The installation program terminates the installation if it finds this variable undefined.

DFC user root directory

DFC creates client-oriented directories (for example, checkout and export) in the user root directory.

On Windows systems, the installation program asks for a user directory root and uses C:\Documentum if you do not specify a location.

On UNIX systems, the installation program uses the environment variable DOCUMENTUM to determine the user directory root. The installation program terminates the installation if it finds this variable undefined.

Directory for shared libraries

The installation program places shared libraries at specific locations relative to the program root directory.

On Windows systems, the installation program uses the shared subdirectory of the program root directory. It attaches the full path of this directory (followed by a separator character) in front of the value of the PATH system environment variable.

On UNIX systems the installation program uses the dfc subdirectory of the program root directory. You must place the full path of this directory onto the library path. The library path environment variable has different names in different versions of UNIX:

- LD_LIBRARY_PATH in Solaris or Linux
- SHLIB_PATH in HP-UX
- LIBPATH in AIX

Environment variables can be set on UNIX systems using the *setenv* script. The script can be found at $DOCUMENTUM_SHARED/dfc/set_dctm_env.sh (.csh). You can source this file to properly set the environment variables from Table 1, page 11.

Directory for DFC configuration files

The installation program creates the config directory to store configuration files. Using the DFC config directory, page 13 provides information about DFC configuration files. The installation program creates the config directory under the program root directory on UNIX systems and under the user root directory on Windows systems. For DFC to operate properly, the full path to the config directory must appear in the classpath.

On Windows systems, the installation program attaches the full path of the config directory (followed by a separator character) in front of the value of the CLASSPATH system environment variable.

On UNIX systems, you must place the full path of the config directory onto the classpath. For example, in the syntax of the csh shell, attach $DOCUMENTUM_SHARED\config: to the value of the CLASSPATH environment variable. You can do this before or after running the installation program, because the installation program does not use this setting.

Locations of DFC classes

The Java runtime environment uses the CLASSPATH environment variable to find DFC classes and the config directory.

On a Windows system, the installation program places the full paths to dctm.jar and the config directory (with appropriate separators) at the front of the classpath.

On a UNIX system the installation program does not modify the classpath. You must place the full paths of dctm.jar and the config directory onto the classpath.

For both Windows and UNIX systems, you must perform an additional step if you want the javac compiler to have access to DFC classes. The javac compiler does not recognize the jars specified in the manifest contained in dctm.jar.

Setting environment variables

DFC uses several environment variables to find its components. Defining file system locations for DFC components, page 9 describes the file system locations that the environment variables point to. On Windows systems, the installation program asks you for the information that it uses to set these variables. On UNIX systems, you must set these variables before you run the installation program. Table 1, page 11 lists these environment variables and summarizes the ways that DFC uses them.

Table 1. Environment variables that DFC uses

Variable	How DFC uses it	Windows value (installation program sets)	UNIX value (you set)
DOCUMENTUM_SHARED	Determine the full path to the program root directory for UNIX	Not used by Windows systems	Specify a value before installing DFC
PATH	Find the directory containing DFC shared libraries (DLLs) on Windows	Attach the full path (followed by a separator character) in front of the *shared* subdirectory of the Documentum program root	Not used by UNIX systems

Variable	How DFC uses it	Windows value (installation program sets)	UNIX value (you set)
Library path (Directory for shared libraries, page 10 lists the different names for this variable on different UNIX systems)	Find the directory containing DFC shared libraries on UNIX	Not used by Windows systems	Add $DOCUMENTUM_SHARED/dfc
DFC_DATA	Documentum has deprecated this variable.	Directory for DFC configuration files, page 10 provides information about what you should do instead of using this variable.	
DOCUMENTUM	Determine the full path to the user root directory	Not used by Windows systems	Specify a value before installing DFC
CLASSPATH	Allow Java runtime to find dctm.jar and, the DFC config directory. See Locations of DFC classes, page 11 for information about making DFC classes available to the javac compiler	Attach (with appropriate separator characters) the full paths of dctm.jar and the config directory (for example, C:\Program Files\Documentum\Shared\dctm.jar and C:\Documentum\config)	Add $DOCUMENTUM_SHARED/dctm.jar and $DOCUMENTUM_SHARED/config

AIX Only

The following sections apply only to AIX installations.

On AIX, DFC is not able to store the registry file on an NFS drive

DFC uses the registry to store/retrieve information for checked out files, viewed files and so on. DFC uses a file-based registry on non-Windows OSs. For a file-based registry, the location of the file is specified explicitly by setting the value of “dfc.registry.file” property in the dfc.properties file. However, if the location is not set, then DFC tries to create the registry file (that is, documentum.ini) in your HOME directory. On AIX machines, if the HOME directory is NFS mounted, DFC cannot access the registry location.

Workaround:

- Specify the registry location in “dfc.properties”. The directory location for the registry file should point to the non-NFS directory. For example:

```
dfc.user.dir=<path to non-NFS dir>
dfc.registry.file=${dfc.user.dir}/documentum.ini
```

- If the registry location is not explicitly specified, DFC tries to create the registry file in your HOME directory. If you prefer this option, your home directory should not be mounted via NFS.

AIX only: Correcting the jaxp.properties file

If you use the IBM J2RE Java runtime environment, you must correct the entries in the jaxp.properties file that pertain to the parser transformation engine. The default location of the jaxp.properties file is in $JAVA_HOME/jre/lib.

The file includes the following property entries:

```
javax.xml.transform.TransformerFactory=
  org.apache.xalan.processor.TransformerFactoryImpl

javax.xml.parsers.SAXParserFactory=
  org.apache.xerces.jaxp.SAXParserFactoryImpl

javax.xml.parsers.DocumentBuilderFactory=
  org.apache.xerces.jaxp.DocumentBuilderFactoryImpl
```

Change these entries to the following:

```
com.documentum.xml.jaxp.DfSAXParserFactory=
  com.documentum.xerces_2_8_0.xerces.jaxp.SAXParserFactoryImpl

com.documentum.xml.jaxp.DfDocumentBuilderFactory=
  com.documentum.xerces_2_8_0.xerces.jaxp.DocumentBuilderFactoryImpl

javax.xml.transform.DfTransformerFactory=
  org.apache.xalan.processor.TransformerFactoryImpl

javax.xml.parsers.DocumentBuilderFactory=
  com.documentum.xerces_2_8_0.xerces.jaxp.DocumentBuilderFactoryImpl

javax.xml.parser.SAXParserFactory=
  com.documentum.xerces_2_8_0.xerces.jaxp.SAXParserFactoryImpl

javax.xml.transform.TransformerFactory=
  org.apache.xalan.processor.TransformerFactoryImpl
```

Using the DFC config directory

The DFC config directory contains Java properties files that control the behavior of DFC. The installation program creates the config directory if it does not already exist. Table 2, page 14 describes the files in the config directory.

Table 2. Configuration files for DFC

File	Description
dfc.properties	Current configuration options for DFC.
dfcfull.properties	Template containing all possible configuration options. Do not modify this file. Copy sections into dfc.properties as necessary.
log4j.properties	Current configuration options for the log4j instance that underlies the unified logging system. *DFC Development Guide* explains the logging system.
dbor.properties	Registry for pre-5.3 business objects. Do not edit this file. *DFC Development Guide* contains information about how to use this file.

Each line of a Java properties file is either a comment (begins with #) or contains a statement of the form *key=value*, where *key* and *value* are character strings that comply with ISO 8859–1 encoding. For characters that do not comply with ISO 8859–1, use Unicode escapes. These are of the form \u followed by the four hexadecimal digits that represent the character's Unicode encoding (for example, \u2297).

The key cannot contain white space. The value can contain spaces and other special characters, but you must precede each with a backslash (\ ) character. For example, to indicate that the DFC configuration files are in C:\Documentum User Files\config, you can include the line:

```
dfc.data.dir=C\:\\Documentum\ User\ Files
```

A backslash precedes each colon, backslash, or space.

The Java convention for expressing file paths allows you to write the same line as:

```
dfc.data.dir=C\:/Documentum\ User\ Files
```

You do not need to precede a forward slash with an escape character.

After installation, the dfc.properties file contains the key dfc.data.dir,. The corresponding value is the full path to the config directory. Directory for DFC configuration files, page 10 provides information about how DFC finds the config directory.

At a minimum, dfc.properties also contains the following keys: dfc.docbroker.host[0], dfc.docbroker.port[0], dfc.tokenstorage.dir, dfc.tokenstorage.enable.

Uninstalling DFC

This section explains how to remove DFC. Whether to remove the old DFC, page 8 lists the situations in which you must remove an old version of DFC before installing the current version.

Regardless of which operating system you use, you cannot uninstall DFC if any program has locked any portion of it. You must stop any program that uses the DFC that you want to uninstall. In particular, this means that you must stop any application server that uses the DFC that you want to uninstall. Stopping an application server terminates any web applications running on it, even those that do not use DFC.

The following sections explain how to uninstall on different operating systems:

- Uninstalling from Windows, page 15
- Uninstalling from UNIX, page 15

Uninstalling from Windows

This section explains how to remove a DFC installation from a Windows system.

If you are uninstalling a DFC version prior to 5.1, refer to Whether to upgrade client programs, page 9 for information about the effect on client programs of upgrading to the current version of DFC.

To uninstall DFC on a Windows system:

1. Change the startup setting to manual for each service that uses DFC.

   This prevents such services (for example, an application server) from restarting themselves after you reboot and locking DFC again before you can uninstall it.
2. Use the control panel's Add/Remove Programs facility to remove Documentum DFC Runtime Environment.
3. If prompted to do so, reboot the system.
4. Restore any startup settings you changed in the first step.

Uninstalling from UNIX

This section explains how to remove a DFC installation from a UNIX system.

If you are uninstalling a DFC version prior to 5.1, refer to Whether to upgrade client programs, page 9 for information about the effect on client programs of upgrading to the current version of DFC.

To uninstall DFC on a Solaris system, run the uninstall.bin program.

To uninstall DFC on a UNIX system other than Solaris, run the uninstall.bin program.

For older versions of DFC, these programs resides in the _uninst subdirectory of the EMC Documentum program root directory. For recent versions, it resides in _uninst/dfc.

Chapter 2

Installing DFC

This chapter describes the process of installing DFC. It contains the following main sections:

Installation requirements

The DFC installation program assumes a video capability of at least 256 colors and at least 800 by 600 screen resolution. For UNIX systems you must also ensure that:

- /usr/dt/bin and /usr/openwin/bin are on the path
- DISPLAY is set to localhost:0.0

Because the installation program provides a graphical interface, you cannot use a telnet session to install DFC. Install from the system console, or use an X server to perform the installation remotely. However, be careful when you install remotely with a DISPLAY setting to localhost:0.0, as the output will be sent to that terminal rather than the one at which you are working.

Installing DFC on a Windows system

This section explains how to install DFC on a Windows system

To install DFC on a Windows system:

1. Run the installation program, dfcWinSuiteSetup.exe.

   The installation program proceeds through a series of dialog boxes and information windows. When necessary, it asks for input.

2. View the Welcome window, and click **Next**.

   The license agreement appears.

3. Select the accept option (**I accept the terms of the license agreement**), and click **Next**.

4. Specify the directory into which the installation program should place the DFC programs and click **Next**.

The default value for this directory is C:\Program Files\Documentum. Refer to Defining file system locations for DFC components, page 9 for information about how DFC uses the value you supply.

The installation program skips this step if it finds a registry entry that contains the required information.

5. Specify whether to install optional components for developers, and click **Next**.

   Select the **Developer Documentation** checkbox to request installation of Javadocs, or leave it unselected if you do not want to have Javadocs installed. The installation program places Javadocs into the help/dfc subdirectory of the DFC program root directory. Open index.html from that subdirectory to view the Javadocs.

6. Specify the root directory for Documentum user information and click **Next**.

   The default value for this directory is C:\Documentum. Refer to Defining file system locations for DFC components, page 9 for information about how DFC uses the value you supply.

   The installation program skips this step if it finds a registry entry that contains the required information.

7. Specify the host and port number for the machine that hosts the connection broker, and click **Next**.

   You can use an IP address or a symbolic address (for example, MyHost.MyCompany.com).

   The installation program skips this step if it finds a dmcl.ini file containing the required information when an upgrade from a Pre-D6 DFC is done. When an upgrade is done from D6 , then it picks the IP from the dfc.properties file from the config directory.

8. Review the summary.

   The installation program summarizes what it plans to install and where it plans to install it. Make a note of anything you want to keep a record of. Use the **Back** button if you want to change anything. Otherwise, click **Next**.

9. Use the checkbox to tell the installation program whether you want to identify the global registry for this DFC to use.

   The installation program skips this step if it finds the required information in the dfc.properties file.

   It is safe to leave the checkbox unselected if you do not have the necessary information. You can set the information manually in the dfc.properties file after the installation program has finished installing DFC.

   If you select the checkbox, the installation program requests the repository and connection credentials, and provides a checkbox you can use to disable validation. Use that checkbox if the repository does not exist or is unavailable. Otherwise, the installation program checks to see if it can use the credentials you provide.

10. Observe the progress of the installation, and click **Finish** when the installation program reports that it has successfully installed DFC.

    The installation program displays progress bars as it completes the installation.

    If the installation program needs to initiate a reboot, it asks before doing so.

Installing on a UNIX system

This section explains how to install DFC on a UNIX system

To install DFC on a UNIX system:

1. Set environment variables, as described in Establishing the environment for DFC, page 9 and Table 1, page 11.
2. Run the installation program.

   The name of the installation program is different for different UNIX systems:

   - dfcSolSuiteSetup.bin for Solaris
   - dfcLinuxSuiteSetup.bin for Linux
   - dfcAixSuiteSetup.bin for AIX
   - dfcHpux11SuiteSetup.bin for HP-UX (32 bit)
   - dfcHpuxia64SuiteSetup.bin for HP-UX (64 bit)

   The installation program proceeds through a series of dialog boxes and information windows. When necessary, it asks for input.
3. View the Welcome window and proceed to the next window.
4. The **license agreement** is displayed.

   Select the accept option (**I accept the terms of the license agreement**), and click **Next**.
5. Specify whether to install optional components for developers, and click **Next**.

   Check the **Developer Documentation** checkbox to request installation of Javadocs, or leave the box unchecked if you do not want to have Javadocs installed. The installation program places Javadocs into the help/dfc subdirectory of the DFC program root directory. Open index.html from that subdirectory to view the Javadocs.
6. Specify the host and port number for the machine that hosts the connection broker, and click Next.

   You can use an IP address or a symbolic address (for example, MyHost.MyCompany.com).

   The installation program skips this step if it finds a dmcl.ini file containing the required information.
7. Review the summary and click **Next** when you are ready to proceed.

   The installation program summarizes what it plans to install and where it plans to install it. Make a note of anything you want to keep a record of. Use the **Back** button if you want to change anything.
8. Use the checkbox to tell the installation program whether you want to identify the global registry for this DFC to use.

   It is safe to leave the checkbox unchecked if you do not have the necessary information. You can set the information manually in the dfc.properties file after the installation program has finished installing DFC.

   If you check the checkbox, the installation program requests the repository and connection credentials, and provides a checkbox you can use to disable validation. Use that checkbox if the repository does not exist or is unavailable. Otherwise, the installation program checks to see if it can use the credentials you provide.

The installation program skips this step if it finds the required information in the dfc.properties file.

9. Observe the progress of the installation, and click **Finish** when the installation program reports that it has successfully installed DFC.

   The installation program displays progress bars as it completes the installation.

10. The installation program replaces copies of the shared library that it finds, but other copies may exist on the machine. It is safe to replace all of them with the current version, but if you do not want to do so, you must ensure that the old version does not precede the current version in any path environment variable that the current DFC might use.

    If the machine has a Content Server installation, you must manually replace the DMCL shared library that is in the server's bin directory.

Chapter 3

Installing Silently

The DFC installation program provides capabilities to support installing silently, that is, invoking the installation program from a command line and giving it a configuration file that allows the installation to proceed without further interaction. This chapter explains how to do this. It contains the following main sections:

Creating the configuration file

To install silently, you must first create a configuration file. To do this, use a command such as the following at a command prompt:

```
dfcWinSuiteSetup.exe -record C:\myFile.properties
```

This is the command for Windows. For other operating systems, use the appropriate executable file rather than dfcWinSuiteSetup.exe. See Step 2 of the installation procedure in Installing on a UNIX system, page 19.

You can replace C:\myFile.properties with any file you choose. Give the full path, not a path relative to the current directory.

Running this command creates myFile.properties as an installer configuration file. It does this by running the installation program interactively and saving your inputs.

Caution: This process records the information during a real-time installation. If you use this method to create your configuration file, it will perform an actual installation during the process.

Running the installation program silently

To run the installation program silently, use a command such as the following at a command prompt:

```
dfcWinSuiteSetup.exe -config C:\myFile.properties -silent
```

Use the same executable file and configuration file as in the previous section.

Because the silent installation program cannot say whether a reboot is required, we recommend always rebooting after a silent installation.

Encrypting passwords

For installation procedures that require users to supply passwords, the installation program provides a rudimentary encryption facility. You can invoke it by using the –x option in the previous sections, as in the following example:

```
dfcWinSuiteSetup.exe -record C:\myFile.properties -x
```

```
dfcWinSuiteSetup.exe -config C:\myFile.properties -silent -x
```

Chapter 4

IPv6 Support

This chapter discusses the IPv6 support.

Documentum client connection process (dual-stack mode)

During startup, Content Server projects its host information such as IP address, port number, and so forth to its connection broker, which maintains the list of all Content Servers to which a client can connect. A Documentum client's DFC retrieves Content Server connection information from the connection broker and stores the IP addresses to which the Content Server listens. The client DFC then connects to Content Server using IPv6 or IPv4. The client DFC first tries to communicate using IPv6. If no connection can be established with IPv6, the client DFC tries IPv4. When the client DFC exhausts both IPv6 and IPv4 connectivity options, an error message is displayed. The client DFC caches the connectivity option for the last successful connection to the Content Server. This improves performance by avoiding repeated attempts to communicate with connection options in which no connectivity can be established. For example, if a client is on a dual stack machine with a connection option set to try_secure_first and the Content Server is listening to an IPv4 non-SSL socket, the client DFC tries to connect to IPv6 SSL, IPv4 SSL, and native IPv6 protocols respectively, before falling back to a native IPv4 connection. The client DFC cache stores the protocol that was successful, and provides it to the client next time the client tries to connect to the Content Server, thus avoiding unnecessary connection attempts.

Note: The DFC cache is refreshed at regular (configurable) time intervals.

Configuring the client DFC

The client DFC retrieves the information about the Content Server by using the full qualified server name (docbase_name.server_config_name@host_name) from the connection brokers listed in its dfc.properties file. In the dfc.properties configuration file, you can specify an IP address or hostname on which the connection broker runs. If you specify an IPv6 address, enclose the IP address in square brackets as per the IPv6 convention.

```
dfc.host.name[0] = [2001:0db8:1234:0000:0000:0000:0000:0000]
```

The following applies when a dual stack client host connects to a connection broker running on a dual stack machine:

- When you specify an IP address in the dfc.properties file, the client DFC uses that IP address and connects without any further processing. For example, if you specify an IPv4 address, the client DFC uses IPv4 for communication.
- When you specify a host name in the dfc.properties, the client DFC resolves all available IP addresses for that host name before determining the connection protocol. When the connection broker runs on a dual stack machine, the client DFC resolves both IPv4 and IPv6 addresses. The client DFC keeps track of the IPv4 address and chooses the best available IPv6 address for the host from Unicast Global, Site Local, and Link Local, in the order specified.

To configure DFC installed on a dual-stack machine for native IPv4 operation, perform the following:

- Specify an IPv4 address in the dfc.properties file.
- Disable the dual-stack operation for Java Virtual Machine.

Configuring the Java Virtual Machine for IPv4 only

A custom property setting in the Java Virtual Machine used by the operating system determines the communications protocol used by the operating system. By default, this custom property (java.net.preferIPv4Stack) is set to False to support dual-stack communications. To configure a host for native IPv4, set this property to True.

Note: On HPUX Java Virtual Machine, the java.net.preferIPv4Stack custom property is not defined. To configure the JVM for dual-stack operation, set the java.net.preferIPv4Stack custom property to False. To configure the JVM for native IPv4, set the java.net.preferIPv4Stack custom property to True.

Chapter 5

Troubleshooting Installer Problems

We implemented this installation program using InstallShield Multiplatform (ISMP).

The installation program maintains an error log, which it writes to a file called setupError.log in the current working directory. If it cannot write into the working directory, it writes to the home directory of the user who initiated the installation. Reading this file may help you see what went wrong. If not, it can help Documentum Technical Support to help you. Everything in this file is important. Send the *entire file, unedited*, to Documentum if you need make a support call. The file does not contain passwords or other secure information.