

EMC[®] Documentum[®]

Documentum Foundation Classes

Version 6.5

Release Notes
P/N 300-007-208-A01

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

Copyright © 2000 - 2008 EMC Corporation. All rights reserved.

Published July 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.

Revision History:

July 2008: Initial Publication.

Table of Contents

Chapter 1	Product Description	7
Chapter 2	New Features and Changes	9
	New features and changes for 6.5	9
	XQuery against XML store	9
	Native 64-bit support	10
	IPv6 support	10
	Documentum Java-Com Bridge (DJCB) is deprecated as of version 6	10
	New features and changes for service pack 6 SP1	10
	Support of the NOT operator	11
	New features and changes for version 6	11
	DMCL API replaced with DFC API	11
	Full format specifications no longer accepted	11
	Character string handling improved	11
	Aspects, a new BOF module type for developers	12
	JMX management of DfPreferences and dfc.properties	12
	New DFC tracing capabilities and configuration settings	12
	DFC deployment	12
	PIA is deprecated in version 6	13
Chapter 3	Fixed Problems	15
	Application_Id is not set properly at startup (131813)	16
	When a user's password expires, there is no way to change it (141927)	16
	Issue with cross-thread IDfPersistentObject operations (144152)	16
	JConsole displays password from dfc.properties as clear text (144936)	16
Chapter 4	Environment and System Requirements	19
	Software requirements	19
	Browser environment	20
	Documentum Foundation Classes environment	29
	Additional operating environments	31
	VMware support	31
	IBM Logical Partitioning support	31
	Citrix support	32
	Native 64-bit support	32
	Prerequisites to run Documentum 6.5 products in the native 64-bit mode	33
	64-bit support in web application server environment	33
	IPv6 support	33
	Cross-product dependencies and interoperability	34
	Embedded products	34
Chapter 5	Known Problems and Limitations	37

Known problems.....	37
DFC validation may not use locale-specific information (53674).....	37
XDQL Query Returns <object> Element With ID = "" If Query Does Not Include r_object_id (62326)	38
XML application does not process object whose name contains "#" (80724).....	38
Exported XML file contains extra entity references (92017)	38
Checkin fails for files linked as peer and child (110216)	38
Two-way repeating attributes fail to update (111976)	39
No registry file on an NFS drive possible (118462).....	39
Excel files in XML format corrupted during import (134069).....	39
Unexpected behavior in IDfCopyOperation (142741)	39
Exception caused while attaching aspect with attributes to an object (143994).....	40
xqueries do not work without default storage (160346, 161027)	40
Problem with XML documents containing a processing instruction or comment after the document end tag (164950)	41
Limitations.....	41
PIA: ByteArrayInputStreamToString does not work with non-ASCII data	42
Java compiler ignores classpath specification in dctm.jar manifest.....	42
DFC does not maintain referential integrity for virtual documents between repositories.....	42
Cannot execute operations within transactions in multi-threaded environment	43
DFC does not support all OLE links.....	43
DFC does not support mixing virtual and other compound documents	43
Can't export an object with multiple formats in a single operation	43
IDfAttr.getDataType and IDfTypedObject.getAttrDataType return unexpected values	43
IDfType.isSubTypeOf returns FALSE if not a direct subtype	44
IDfTime.equals returns true if time difference is less than one second	44
IDfPersistentObject.getChildRelatives ignores setting of child_value	44
Importing Windows folder containing a shortcut does not import the file or folder the shortcut refers to	44
Importing 'in progress' XML document fails if DDS validation is ON	45
XML application cannot process xml:base attributes	45
Transformations lose whitespace	45
Can't check out replica, then check in original	45
Can't use two repositories with the same name.....	46
File download dialog box does not appear during http transfer operations.....	46
Problems with importing a document on Macintosh platform	46
Replication of objects with aspects is not supported	47
Problems with netegrity multidocbase	47
Not possible to prevent renditions from deletion	47
Issue when downgrading DFC	47
Export operation of folder with special characters fails.....	48
Chapter 6 Technical Notes	49
Required configuration for machine-only application access control tokens.....	49
Ensuring a correct jaxp.properties File	49

	Setting permissions for multiple DFC users on Unix systems.....	50
	Maximum number of results to retrieve per source by a query search	51
	Changing the dfc.properties dynamically on Windows	51
	Copying a keystore from a machine to another does not work	51
Chapter 7	Documentation	53
	Documentum Foundation Classes documentation set	53
	Obtaining the correct documentation	54
	Documentation corrections and clarifications	54
	Aspect deployment.....	54
	DMCL/DFC compatibility issues	54
	String values no longer truncated by default	54
	Routers no longer supported	55
	Nameless getFile()	55
	Failed save() or checkin() operations	55
	Compatibility preferences	55
	setString()	57
	setDocbaseScope() method deprecated.....	58
	Session management enhancements.....	58
	DFC Development Guide: Documentation of do- method signatures	59
Chapter 8	Software Media, Organization, and Files	61
	Software Media	61
	Organization	61
	Files.....	61
Chapter 9	Installation	63
Chapter 10	Troubleshooting and Getting Help	65

List of Tables

Table 1.	Browser environment	21
Table 2.	Documentum Foundation Classes environment.....	29
Table 3.	VMware support	31
Table 4.	IBM Logical Partitioning support.....	31
Table 5.	Citrix support.....	32
Table 6.	Documentum Foundation Classes IPv6 compliance	34

Product Description

EMC Documentum functionality has the following main elements:

Repositories	One or more places where you keep the content and associated metadata of your organization's information. The metadata resides in a relational database, and the content resides in various storage elements.
Content Server	Software that manages, protects, and imposes an object oriented structure on the information in repositories. It provides tools for managing the lifecycles of that information and automating processes for manipulating it.
Client programs	Software that provides interfaces between Content Server and end users. The most common clients run on application servers (for example, Webtop).
End Users	People who control, contribute, or use your organization's information. They use a browser to access client programs running on application servers.

Documentum Foundation Classes (DFC) lies between Content Server and clients. While the main user of DFC is other EMC Documentum software, you can use DFC in any of the following ways:

- Access EMC Documentum functionality from within one of your company's enterprise applications.
- Customize or extend products such as Webtop.
- Write a method or procedure for Content Server to execute as part of a workflow or document lifecycle.

DFC is Java based. As a result, client programs that are Java based can interface directly with DFC.

DFC is the basis for the business object framework (BOF), which is the principal tool for adding custom business logic to EMC Documentum clients (for example Webtop). BOF enables you to embody business rules and patterns in reusable elements, called modules. BOF makes it possible to extend some of DFC's implementation classes. As a result, you can introduce new functionality in such a way that unmodified existing programs begin immediately to deliver the new functionality.

DFC and BOF are the basis for the web services framework (WSF), a framework for delivering EMC Documentum functionality as web services.

DFC runs on a Java virtual machine (JVM), which can be on:

- A middle-tier system.
For example, on an application server to support WDK or to execute server methods.
- The machine that runs Content Server.
- An end user's computer.

New Features and Changes

EMC Documentum Foundation Classes version 6.5 includes a variety of new features and improvements. This section describes the major enhancements. Refer to the product documentation for complete information about these features.

New features and changes for 6.5

This section lists new features and changes for 6.5.

XQuery against XML store

This release of DFC provides a new interface (IDfXQuery) for issuing XQueries against the XML store. With XQueries, you can address any piece of information in an XML document, make selections based on conditions, and construct new structures based on query result sets. Query results return XML documents or byte streams that adhere to Documentum security.

This release adds a new extension function, that can be invoked through XQuery, which returns DQL query results as an XML document. Using XQuery, you can join data (from the relational database) returned by DQL queries with the data or metadata extracted from the XML store.

The XQuery interface can be accessed through DFC and IAPI.

The following illustrates a couple of common use cases:

- Search use case – find all documents that contain a particular XML tag and one or more Documentum attribute values.
- Constrained query use case – find a document or set of documents based on a Documentum attribute value (such as project name or folder path).

- Metadata results use case – construct a results document comprised of content from a set of XML documents that matches the search criteria, and include Documentum attribute values associated with those XML documents in the constructed document.

Note: XQuery cannot be used to update, insert, or delete content in the XML store.

For more information on XQuery, see <http://www.w3.org/TR/xquery/>.

Native 64-bit support

Documentum 6.5 introduces native 64-bit support for running on 64-bit operating systems. The native 64-bit support takes full advantage of 64-bits of address space and the concurrent use of more than 4 GB of executable memory. For a fully qualified native 64-bit environment, you must have a 64-bit processor, 64-bit operating system, 64-bit JVM, and 64-bit application server. Support continues to be offered on 32-bit platforms.

IPv6 support

This product is compliant with IPv4-IPv6 dual-stack and native IPv6 enabled environments. The Environment and System Requirements chapter provides more information on IPv6 support for the product.

Documentum Java-Com Bridge (DJCB) is deprecated as of version 6

Documentum Java-Com Bridge (DJCB) is deprecated as of version 6.

New features and changes for service pack 6 SP1

This section lists new features and changes for service pack 6 SP1.

Support of the NOT operator

Full-text queries, that can be run from the simple search box or from the Contains box in the advanced search page, now support the NOT operator. The NOT operator allows to exclude the term it precedes from the query. It can be used alone at the beginning of the query, between two terms or with the AND or OR operators. Some limitations on external sources are documented in the user documentation.

New features and changes for version 6

This section lists new features and changes for service pack 6.

DMCL API replaced with DFC API

The C++ DMCL API has been replaced with the Java-based DFC API. These core changes, while extremely significant, are largely transparent to the DFC user. C++ applications that interact directly with the DMCL continue to work as a copy of DMCL continues to be provided. New Documentum 6 features are not available through DMCL, however.

Full format specifications no longer accepted

DFC methods, such as `setFile`, that previously accepted a full format specification no longer do so. Those methods accept only a format name, such as `txt` or `word`, for the format argument.

Character string handling improved

In previous releases, if you attempted to set a character string property with a value that exceeded the defined length of the property, DFC quietly truncated the value to the maximum length of the property and then set the property. For Documentum 6, DFC throws an exception instead of truncating the value and setting the property. If you prefer to use the pre Documentum 6 behavior, set the `dfc.compatibility.truncate_long_values` property in `dfc.properties` file to `T`. This property is `false` by default.

Aspects, a new BOF module type for developers

Documentum 6 supports aspects, a new framework for extending object behavior and attributes. Aspects are a type of BOF entity that can be dynamically attached to object instances, to provide fields and methods beyond the standard ones for the object type. The extended behavior can include functionality that applies to types across the object hierarchy; for example, an aspect could label objects as retainable or web-viewable, and this single aspect could be applied to multiple distinct object types. Using aspects can speed development and improve code reuse, because the extended attributes and behavior do not alter the underlying type definitions.

You can create aspects and associate them with an individual object or an object type. If you associate them with an object type, the aspect is automatically associated with each new object of the specified object type.

Aspects can also have properties defined for them. Properties defined for an aspect appear to users as if they are defined for the object type of the object to which the aspect is attached.

JMX management of DfPreferences and dfc.properties

In J2EE DFC-based applications, active settings in DfPreferences and persistent settings in dfc.properties are managed by JMX agent and Managed Bean (MBean) components. The settings are displayed in Documentum Administrator, which separates active settings (in DfPreferences) from persistent settings (in dfc.properties).

New DFC tracing capabilities and configuration settings

Tracing is now controlled by entries in the dfc.properties file, rather than the log4j file. Additionally, this release adds multiple new tracing capabilities. For example, you can now set up tracing to trace by user, by thread, and call stack depth. You can also define the format of the tracing file.

DFC deployment

DFC is deployed with each application or product that requires it, using a standard J2EE deployment strategy. In the new deployment process, the dfc.jar file and related files are packaged in a product's WAR file so that each DFC instance can have its own DFC configuration.

This means that your own applications can also be deployed in such a way that they have their own sandboxed copy of the `dfc.jar`, minimizing the potential for conflicts between your applications and other processes running on your Documentum server implementation.

PIA is deprecated in version 6

PIA is deprecated in 6 release.

Fixed Problems

Within two weeks of the release, a list of the fixed problems in the release will be posted on the Powerlink website (<http://Powerlink.EMC.com>)

To view the list of fixed problems:

1. Log in to the Powerlink website.
You must have a software support agreement to log in and access the list of fixed bugs.
2. From the Powerlink menu bar, select **Support > Knowledgebase Search > Documentation and White Papers Search**.
The Knowledgebase Search screen appears.
3. In the **Search For** text box, enter `Fixed`.
4. From the **Select Document Type(s)** drop-down list, select `FIXED BUGS LIST`.
5. Select the product whose fixed problems you want to view.
 - a. Select `Software` from the **Product Type** drop-down list.
 - b. Select `Content` from the **Product Family** drop-down list.
 - c. Select the product name from the **Product** drop-down list.
 - d. Select the product version number from the **Version** drop-down list.
6. Click the **Search** button.
7. Scroll to the bottom of the page and select the fixed problems document from the list of search results.

Application_Id is not set properly at startup (131813)

The Application_Id variable is not set on sessions obtained from a pool.

Workaround: Set the following in dfc.properties.

```
dfc.session.pool.mode=level1
```

When a user's password expires, there is no way to change it (141927)

When a user's OS password expires, there is currently no way of changing it using IDfClient.authenticate(). Previously, you were able to get an IDfSession that contained an error message stating that the password had expired, but that you could still use the password to change the password.

Issue with cross-thread IDfPersistentObject operations (144152)

If an IDfPersistentObject is fetched in one Java thread (T1), and passed on to another Java thread (T2), and T2 is running a getType() operation using the READ_QUERY flag while T1 performs a commit operation such as an update, then T2's database cursor closes, returning an incorrect null result for getType() on a properly fetched IDfPersistentObject.

JConsole displays password from dfc.properties as clear text (144936)

If you have a WDK-based product WAR file expanded on the application server file system, and start the application server with a script that has been amended to configure JConsole, the decrypted password is displayed in the Attributes tab. If you enter a new

password, it is displayed as clear text. If you save the attribute settings, the password is saved as clear text (not encrypted) in the file on the application server file system.

Environment and System Requirements

This chapter lists the hardware (machine) requirements and software versions supported with this release. Machine resources, non-Documentum software components (Operating System, Java Runtime Environment, and so forth), and other EMC Documentum products determine the unique environment for each EMC Documentum product. , addresses basic machine resource requirements. [Software requirements, page 19](#), addresses specific software versions that are required for the installation platform.

All EMC Documentum products use Documentum Foundation Classes (DFC). Many deploy specific versions of Documentum Foundation Classes. In some cases, these products do not work properly with the version of Documentum Foundation Classes required by other EMC Documentum products. In general, all products with the same version number (Documentum 6, for example) use the same version of Documentum Foundation Classes.

Documentum Foundation Classes maintains backward compatibility. In general, deploying a newer version of Documentum Foundation Classes does not cause older products to malfunction. The opposite, however, is not true. Replacing a newer Documentum Foundation Classes version with an older one may cause products that depend on the newer version to malfunction.

Documentum Foundation Classes operates in a Java Runtime Environment (JRE). In general, different programs on the same machine can use different versions of Documentum Foundation Classes running in different JREs. A web application server, for example, might use its own version of Documentum Foundation Classes. In some environments, such as Windows, however, you cannot have more than one version of Documentum Foundation Classes installed (refer to [Documentum Foundation Classes environment, page 29](#)).

Software requirements

This section provides information on supported software environments.

The tables in this section reflect the latest versions of third-party products, upon which the EMC Documentum product depends, that are supported at the time of this release. For information on currently supported environments and future updates, refer to

Product Information System on the Powerlink website: <http://Powerlink.EMC.com>.
For detailed information about the policies governing supported products, refer to the *Customer Guide to EMC Software Support Services* available on the Powerlink website: <http://Powerlink.EMC.com>.

Browser environment

Unified Client Facilities content transfer is supported with Documentum Foundation Classes for the following browser environments.

Table 1. Browser environment

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Windows XP SP2 (32-bit version)	IA-32, x64	Sun JRE 1.4.2_16		Sun JRE 1.4.2_16		Sun JRE 1.4.2_16		
Windows XP SP2 (32-bit version)	IA-32, x64	Sun JRE 5.0 Update 14		Sun JRE 5.0 Update 14		Sun JRE 5.0 Update 14		
Windows XP SP2 (32-bit version)	IA-32, x64	Sun JRE 6.0 Update 5		Sun JRE 6.0 Update 5		Sun JRE 6.0 Update 5		
Windows Vista SP1 (32-bit version)	IA-32, x64				Sun JRE 1.4.2_16	Sun JRE 1.4.2_16		
Windows Vista SP1 (32-bit version)	IA-32, x64				Sun JRE 5.0 Update 14	Sun JRE 5.0 Update 14		
Windows Vista SP1 (32-bit version)	IA-32, x64				Sun JRE 6.0 Update 5	Sun JRE 6.0 Update 5		

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Windows Server 2003 R2 with SP2 (32-bit version)	IA-32, x64		Sun JRE 1.4.2_16	Sun JRE 1.4.2_16		Sun JRE 1.4.2_16		
Windows Server 2003 R2 with SP2 (32-bit version)	IA-32, x64		Sun JRE 5.0 Update 14	Sun JRE 5.0 Update 14		Sun JRE 5.0 Update 14		
Windows Server 2003 R2 with SP2 (32-bit version)	IA-32, x64		Sun JRE 6.0 Update 5	Sun JRE 6.0 Update 5		Sun JRE 6.0 Update 5		
Windows Server 2003 SP2 (32-bit version)	IA-32, x64		Sun JRE 1.4.2_16	Sun JRE 1.4.2_16		Sun JRE 1.4.2_16		
Windows Server 2003 SP2 (32-bit version)	IA-32, x64		Sun JRE 5.0 Update 14	Sun JRE 5.0 Update 14		Sun JRE 5.0 Update 14		
Windows Server 2003 SP2 (32-bit version)	IA-32, x64		Sun JRE 6.0 Update 5	Sun JRE 6.0 Update 5		Sun JRE 6.0 Update 5		

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Mac OS X 10.4.10	PowerPC					Java for Mac OS X 10.4 Release 6 (1.4.2_16)	Java for Mac OS X 10.4 Release 6 (1.4.2_16)	
Mac OS X 10.4.10	PowerPC					Java for Mac OS X 10.4 Release 6 (1.5.0_13)	Java for Mac OS X 10.4 Release 6 (1.5.0_13)	
Mac OS X 10.4.10	IA-32, x64					Java for Mac OS X 10.4 Release 6 (1.4.2_16)	Java for Mac OS X 10.4 Release 6 (1.4.2_16)	

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Mac OS X 10.4.10	IA-32, x64					Java for Mac OS X 10.4 Release 6 (1.5.0_13)	Java for Mac OS X 10.4 Release 6 (1.5.0_13)	
Mac OS X 10.4.11	PowerPC					Java for Mac OS X 10.4 Release 6 (1.4.2_16)		Java for Mac OS X 10.4 Release 6 (1.4.2_16)
Mac OS X 10.4.11	PowerPC					Java for Mac OS X 10.4 Release 6 (1.5.0_13)		Java for Mac OS X 10.4 Release 6 (1.5.0_13)

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Mac OS X 10.4.11	IA-32, x64					Java for Mac OS X 10.4 Release 6 (1.4.2_16)		Java for Mac OS X 10.4 Release 6 (1.4.2_16)
Mac OS X 10.4.11	IA-32, x64					Java for Mac OS X 10.4 Release 6 (1.5.0_13)		Java for Mac OS X 10.4 Release 6 (1.5.0_13)
Mac OS X 10.5.3	PowerPC					Java for Mac OS X 10.5 (1.4.2_16)		Java for Mac OS X 10.5 (1.4.2_16)
Mac OS X 10.5.3	PowerPC					Java for Mac OS X 10.5 (1.5.0_13)		Java for Mac OS X 10.5 (1.5.0_13)

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Mac OS X 10.5.3	IA-32, x64					Java for Mac OS X 10.5 (1.4.2_16)		Java for Mac OS X 10.5 (1.4.2_16)
Mac OS X 10.5.3	IA-32, x64					Java for Mac OS X 10.5 (1.5.0_13)		Java for Mac OS X 10.5 (1.5.0_13)
Solaris 10	Ultra-SPARC					Sun JRE 1.4.2_16		
Solaris 10	Ultra-SPARC					Sun JRE 5.0 Update 14		
Solaris 10	Ultra-SPARC					Sun JRE 6.0 Update 5		
Red Hat Enterprise Linux 4.6	IA-32, x64 [1]					Sun JRE 1.4.2_16		

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
Red Hat Enterprise Linux 4.6	IA-32, x64 [1]					Sun JRE 5.0 Update 14		
Red Hat Enterprise Linux 4.6	IA-32, x64 [1]					Sun JRE 6.0 Update 5		
Red Hat Enterprise Linux 5.1	IA-32, x64 [1]					Sun JRE 1.4.2_16		
Red Hat Enterprise Linux 5.1	IA-32, x64 [1]					Sun JRE 5.0 Update 14		
Red Hat Enterprise Linux 5.1	IA-32, x64 [1]					Sun JRE 6.0 Update 5		
SUSE Linux Enterprise Desktop 10 SP1	IA-32, x64 [1]					Sun JRE 1.4.2_17		

		Browser & Java Plug-In						
Operating system	Processor	Internet Explorer 6 for Windows XP SP2 (6.00.2900.2180)	Internet Explorer 6 for Windows Server 2003 (6.00.3790.0000)	Internet Explorer 7 for Windows XP and Windows Server 2003 (7.00.5730.1100)	Internet Explorer 7 for Windows Vista (7.00.6000.16386)	Firefox 2.0.0.12	Safari 2.0.4	Safari 3.0.4
SUSE Linux Enterprise Desktop 10 SP1	IA-32, x64 [1]					Sun JRE 5.0 Update 14		
SUSE Linux Enterprise Desktop 10 SP1	IA-32, x64 [1]					Sun JRE 6.0 Update 5		
SUSE Linux Enterprise Server 10 SP1	IA-32, x64 [1]					Sun JRE 1.4.2_16		
SUSE Linux Enterprise Server 10 SP1	IA-32, x64 [1]					Sun JRE 5.0 Update 14		
SUSE Linux Enterprise Server 10 SP1	IA-32, x64 [1]					Sun JRE 6.0 Update 4		

Note: The following specific notes indicated by [] apply to the Browser environment table:

[1] Support 64-bit version of operating system through the 32-bit compatibility mode.

Note: The following general notes apply to the Browser environment table:

- The following Windows Server 2003 editions are supported: Standard, Enterprise, Data Center.
- IA-32 = Intel Architecture 32-bit
- x64 = 64-bit Architecture which includes AMD64 and Intel 64
- For Windows XP SP2, you must either disable its built-in popup-blocker, or you must add the web application server as a trusted site.
- Cookies must be enabled in the browser.
- In order for UCF content transfer related operations such as import, export, view, checkout, checkin, preferences to work under the Windows Vista operating system, you must choose one of the following: turn off User Account Control (UAC) for each user, or in the browser add the site to the list of Trusted Sites in IE7 and in the application set the UCF client configuration registry.mode value to file instead of windows.
- The following Red Hat Enterprise Linux editions are supported: AS, ES, WS, and Desktop.
- Tier 1 language versions of the operating system and browser are supported. Tier 1 languages are French, Italian, German, Spanish, Japanese, Korean, Simplified Chinese, Swedish, Russian, and Brazilian Portuguese.

Documentum Foundation Classes environment

The basic software platform for Documentum Foundation Classes consists of an operating system and a Java Runtime Environment (JRE). The following table lists the supported platforms for Documentum Foundation Classes.

Table 2. Documentum Foundation Classes environment

Operating system	Processor
Windows XP SP3 (32-bit version)	IA-32, x64
Windows Vista SP1 (32-bit version)	IA-32, x64
Windows Server 2003 R2 with SP2 (32-bit version)	IA-32, x64
Windows Server 2003 SP2 (32-bit version)	IA-32, x64
Windows Server 2003 R2 with SP2 x64 Edition	x64 [2]

Operating system	Processor
Windows Server 2003 SP2 x64 Edition	x64 [2]
Solaris 10	UltraSPARC
AIX 5L V5.3 TL7	Power
AIX 5L V6.1	Power
HP-UX 11i v2 Update 2 (B.11.23)	PA-RISC 2.0
HP-UX 11i v2 Update 2 (B.11.23)	Itanium 2
HP-UX 11i v3 Update 1 (B.11.31)	PA-RISC 2.0
HP-UX 11i v3 Update 1 (B.11.31)	Itanium 2
Red Hat Enterprise Linux 4.6	IA-32, x64 [1]
Red Hat Enterprise Linux 4.6	x64 [2]
Red Hat Enterprise Linux 5.1	IA-32, x64 [1]
Red Hat Enterprise Linux 5.1	x64 [2]
SUSE Linux Enterprise Server 10 SP1	IA-32, x64 [1]
SUSE Linux Enterprise Server 10 SP1	x64 [2]

Note: The following specific note indicated by [] applies to the Documentum Foundation Classes environment table:

- [1] Support 64-bit version of operating system through the 32-bit compatibility mode.
- [2] Support 64-bit version of operating system in native mode.

Note: The following general notes apply to the Documentum Foundation Classes environment table:

- The following Windows Server 2003 editions are supported: Standard, Enterprise, Data Center.
- IA-32 = Intel Architecture 32-bit
- x64 = 64-bit Architecture which includes AMD64 and Intel 64
- For Documentum Foundation Classes in standalone mode, Java J2SE 5.0.x is supported. For Documentum Foundation Classes running in a web application server environment, all Java 5.0.x versions supported by each web application server configuration are supported. No support for Java 1.4.x.
- For 32-bit operating system, Documentum Foundation Classes is supported with 32-bit Java only. For 64-bit operating system, Documentum Foundation Classes is supported with 32-bit Java (32-bit compatibility mode) and 64-bit Java (native).
- Support 32-bit version and 64-bit version of application server.
- English, Japanese, Korean and Simplified Chinese versions of the operating system are supported.

Additional operating environments

This section provides information on supported operating environments.

VMware support

The following table lists the VMware product versions supported on specific operating system(s).

Table 3. VMware support

Product	Operating system	Processor
VMware Server	Windows & Linux (See VMware website for the specific supported version)	IA-32, x64
VMware Workstation	Windows & Linux (See VMware website for the specific supported version)	IA-32, x64
VMware ESX Server	N/A	IA-32, x64

Note: The following general notes apply to the VMware support table:

- IA-32 = Intel Architecture 32-bit
- x64 = 64-bit Architecture which includes AMD64 and Intel 64
- Documentum fully supports all of our products other than index server running on a guest operation system supported by the version of VMware products listed as long as that same version of operating system is supported by the appropriate EMC Documentum product. VMware supports only x86 microprocessor architecture.

IBM Logical Partitioning support

The following table lists the IBM Logical Partitioning (LPAR) product versions supported on specific operating system(s).

Table 4. IBM Logical Partitioning support

Product versions	Operating system	Processor
IBM Logical Partitioning (LPAR) [1] [2]	AIX 5L V5.3 TL7 AIX 5L V6.1	Power

Note: The following specific note indicated by [] applies to the IBM Logical Partitioning support table:

- [1] IBM Logical Partitioning is not supported for Index Server.
- [2] At this time, recommendations regarding performance tuning and configuration shall come from IBM customer representatives, business partner and/or IBM best practice documents for deploying LPARs. EMC will attempt to reproduce customer problems by running in a non-LPAR environment. Should the problem be related to running in a LPAR environment, EMC will collaborate with IBM in correcting the problem. We expect the customer to perform relevant use case scenarios pre-production testing on LPARS. This limited support assumes that these tests and the configurations are performed successfully in a test environment prior to going into production.

Citrix support

The following table lists the Citrix product versions supported on specific operating system(s).

Table 5. Citrix support

Product versions	Operating system	Processor
Citrix Presentation Server 4.5 FP1 [1]	Windows Server 2003 SP2 (32-bit version)	IA-32, x64

Note: The following specific note indicated by [] applies to the IBM Logical Partitioning support table:

- [1] Not supported for Documentum Application Connectors.

Note: The following general notes apply to the Citrix support table:

- IA-32 = Intel Architecture 32-bit
- x64 = 64-bit Architecture which includes AMD64 and Intel 64

Native 64-bit support

Documentum 6.5 introduces native 64-bit support such that 64-bit Documentum applications can run on 64-bit operating systems.

Prerequisites to run Documentum 6.5 products in the native 64-bit mode

You must ensure that your 64-bit environment comprises the following prerequisite components before you run the Documentum 6.5 products:

- 64-bit processor
- 64-bit operating system
- 64-bit Java Virtual Machine
- 64-bit web application server (if needed). Ensure that the web application server is bundled with 64-bit Java Virtual Machine.

64-bit support in web application server environment

Documentum 6.5 continues to support 32-bit versions of the Microsoft Windows operating system. In the 64-bit domain, Documentum 6.5 does not support 64-bit versions of the Microsoft Windows operating system. However, as an exception, Documentum 6.5 provides native support of the 64-bit version of Microsoft Windows Server 2003 for web application server- and portal server-based products.

Note: The following general notes apply to Native 64-bit support in Documentum 6.5:

- Check your web application server documentation for information about configuration changes you need to make to run Documentum 6.5 product on a 64-bit platform. For example, the IBM WebSphere Application Server V6.1 Fix Pack 13 (6.1.0.13) ships with platform-specific 32-bit and 64-bit installers.
- For information about tuning a 64-bit Java Virtual Machine and garbage collection parameters, refer to the Java Virtual Machine documents of the relevant vendor.
- Porting the 32-bit Documentum 6.5 application to a 64-bit platform: Documentum 6.5 product must install a 64-bit compatible Java Virtual Machine before porting the 32-bit Documentum 6.5 application to a 64-bit platform.

IPv6 support

Documentum 6.5 now supports dual-stack (IPv4 or IPv6) and native IPv4 operation. In addition to this, Documentum 6.5 products which are Java based applications also support native IPv6 operation. Support for dual-stack operation enables Documentum 6.5 products to use the appropriate protocol (IPv4 or IPv6) when communicating with third-party products and other Documentum products.

Documentum Foundation Classes determines the operating mode, and by default, DFC is configured for dual-stack and native IPv6 operation. The *Documentum Foundation Classes Installation Guide* provides information on how to configure DFC for native IPv4 operation.

Note: Information on protocol support for third-party products is provided in the third-party product documentation.

Table 6. Documentum Foundation Classes IPv6 compliance

Product	Dual-Stack (IPv4 & IPv6)	Native IPv6
Documentum Foundation Classes	X	X
Note: The X symbol denotes the support started with 6.5 release.		

Cross-product dependencies and interoperability

This section lists Documentum products that are depended on in order to achieve a basic level of functionality and lists related Documentum products that enhance overall functionality.

You may have to install some of the listed products on separate host or client machines due to differences in the Documentum Foundation Classes versions included with those products. Before installing a product, check the product's *Release Notes* for supported configurations.

The 32-bit version of Documentum 6.5 products can run in the 32-bit native mode and the 64-bit compatibility mode. However, the 64-bit version of Documentum 6.5 products can run only in the 64-bit native mode.

For Documentum Foundation Classes, the following products are prerequisites:

- Content Server 5.3, 5.3 SPx, 6, 6 SP1, 6.5

Note: 5.3 global registry is not supported.

Embedded products

Installing EMC Documentum products includes the following additional products.

For Documentum Foundation Classes:

- Xerces 2.8.0
- Xalan 2.7.0
- JXPath 1.2
- AspectJ 1.5.2a

- Log4j 1.2.13

Known Problems and Limitations

This section identifies problems and limitations that may affect your use of the product.

Note: This section and the Technical Notes section may refer to platforms or features that are not supported for this release of your product. Check [Chapter 4, Environment and System Requirements](#) to verify requirements.

EMC Documentum makes the latest information about customer-reported issues and known problems are posted on the Powerlink website (<http://Powerlink.EMC.com>). You must have a software support agreement to log in and access the list of issues.

Known problems

This section describes known defects in EMC Documentum software that may affect your use of the product.

DFC validation may not use locale-specific information (53674)

DFC validation methods obtain the locale from `sessionconfig.session_locale` and use that locale in data dictionary queries. If the data dictionary does not have information for that locale, the queries return an empty result set, rather than information for the most closely matching locale.

Workaround: Before using DFC validation methods, ensure that `sessionconfig.session_locale` contains a locale for which the data dictionary contains locale-specific values.

XDQL Query Returns <object> Element With ID = "" If Query Does Not Include r_object_id (62326)

If you submit an XDQL query that does not include `r_object_id`, as, for example,

```
query.setDql("select object_name from dm_cabinet");
query.execute(DfXmlQuery.DF_READ_QUERY, sess);
org.w3c.dom.Document document = query.getXMLDOM();
String str = query.getXMLString();
System.out.println(str);
```

the returned XML has the following form:

```
<?xml version="1.0"?>
<root>
  <object ID="">
    <object_name>s28o426</object_name>
  </object>
```

Workaround: Process the output to remove the unwanted `ID=""`.

XML application does not process object whose name contains "#" (80724)

You can check out an XML object whose name contains "#," but the XML application does not process the object if you try to check it in.

Workaround: Do not use "#" in the name of an XML object.

Exported XML file contains extra entity references (92017)

When DFC exports an object from a repository, the file contains entity references that are already contained in the DTD.

Checkin fails for files linked as peer and child (110216)

If you check in an XML document that has file links (child links) and peer links to the same files, DFC does not check in the linked files.

Two-way repeating attributes fail to update (111976)

If an XML application has a two-way repeating attribute, and if you change the value of the attribute in the repository and check out the XML document, the attributes still have their original values in the local document.

No registry file on an NFS drive possible (118462)

DFC uses the registry to store/retrieve information for checked out files, viewed files and so on. By default, DFC uses Windows registry on Windows OS and file-based registry on non-Windows OS. For file-based registry, the location of the file can be 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 then DFC cannot access the registry location.

Workaround:

- You have to explicitly 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 then DFC tries to create the registry file in your HOME directory. If this option is preferred then your home directory should be non-NFS mounted.

Excel files in XML format corrupted during import (134069)

Excel files saved in XML format are corrupted during processing by the default XML application. The namespace xmlns="urn:schemas-microsoft-com:office:office" is removed from the Workbook section. The XML file then cannot be opened by Excel.

Unexpected behavior in IDfCopyOperation (142741)

When the copy preference for IDfCopyOperation is set to 1 (COPY_REFERENCE) and child_copy is set to 0 (zero) on the dmr_containment, behavior when copying a virtual

document is always 2 (COPY_COPY): it copies the children, when it should reference them.

Exception caused while attaching aspect with attributes to an object (143994)

When you try to attach aspect with attributes to an object, an exception occurs.

xqueries do not work without default storage (160346, 161027)

The problem relate to failures occurring when storing xml content into the repository. This happens when the XML store feature is enabled and the format object for XML does not have its default_store value set to an XML Store. Xqueries executed under this scenario will result results in a [DM_STORAGE_E_INVALID_OBJECT_ID]error: "0000000000000000 is not a valid external store object id" error.

Workaround: You can use either of the following to avoid encountering this issue:

1. Set the default_store of the XML format object to the XML Store. This will force all xqueries to run against the XML Store specified in the format object. However, this could pose a problem for applications using older versions of DFC (pre-6.5) as they will be unable to create or update XML Content since older versions of DFC do not support writing to an External Store.
2. You can specify at least one xml store to query against when running an xquery. This is done by calling the DFC IDfXQuery.addStore method. If at least one xml store is specified, then the Content Server will use the specified store instead of using the value set in the default_storage attribute of the xml format object. This workaround does not work for running xqueries using the IAPI xquery command, as there is no option to specify a store.

Example 5-1.

```
ISession s = (ISession)sm.newSession(docbaseName);
IDfClientX clientx = new DfClientX();
IDfXQuery xquery = clientx.getXQuery();
xquery.setXQueryString("for $article in /chapter[@number=4]//article
\r\n return <li>>{data($article/para)}</li>");
xquery.addStore("testenv_xmlstore");
xquery.execute(s);
```

Problem with XML documents containing a processing instruction or comment after the document end tag (164950)

The processing instruction at the end of an XML document causes the document to be saved in a corrupt state in the repository. The document cannot be exported or checked out.

Workaround: Remove the processing instruction or comment that follows the document end tag. This will most likely be encountered when you use the Epic XML editor.

Example 5-2.

```
<?xml version=="1.0" encoding="UTF-8"?>
<!DOCTYPE book PUBLIC "-//Arbortext//DTD DocBook XML V4.0//EN" "file:
C:\Documentum\XML%20Applications\dm_notes\DCTM%20TechPubs\axdocbook.dtd">
<book>
...
</book>
<?Pub *0000979793 0?>
```

This sort of document will import (or checkin) successfully, however, on export or checkout, a variety of errors can occur. One sort of error can occur if an entity is defined in the DTD and is used in the XML document. For example, if you use the – entity, the export will fail with a message like this:

```
Operation failed: The entity "ndash" was referenced, but not declared. Line 380,
column 35 ,file:/tmp/documentum/contentXfer/appserver-2008.07.08-1720h.11s_43149/
1d521ab1861q117c7e02c621q1d391/dctm_filename_8_3877.xml; Additional msg - object
name: Document Name
```

Limitations

This section describes limits on the usability of current functionality. The limitations may be part of the product design or may result from issues with associated third-party products.

Architectural and design decisions cause DFC to behave in certain ways and prevent it from behaving in others. Design limitations arise when such decisions preclude behaviors that might otherwise be desirable. In some cases EMC Documentum cannot change behaviors because they occur in public interfaces. These are also design limitations.

Each of the following sections describes a design limitation. These are aspects of the product that are not likely to change.

PIA: ByteArrayInputStreamToString does not work with non-ASCII data

Using PIA, the client is unable to use the `getContent()` method to read data into memory. The recommended approach for reading the data into memory is to use a code snippet such as the following:

```
dfclientX.ByteArrayInputStreamToString (sysObject.getContent())
```

(The return value of `getContent()` is not a defined class in C#/.NET.) This approach works for ASCII data. For non-ASCII data, only a small portion of the data is read .

Workaround: Use `getFile(filename)`. This workaround has performance and security considerations.

Java compiler ignores classpath specification in dctm.jar manifest

The Sun Java compiler (`javac`) ignores the classpath specifications in the manifest of the `dctm.jar` file. Sun regards this as the correct behavior, so it is not likely to change.

Workaround: Add `dfc.jar` (and other jars from the manifest, if necessary) to your classpath to compile Java files that import DFC packages or classes.

Documentum WDK performs this workaround in the scripts it uses to start the application server, so this problem does not exist in the WDK environment.

DFC does not maintain referential integrity for virtual documents between repositories

DFC does not check, before deleting a document, to see if that document is a child of a virtual document in another repository. For example, assume Doc A, in repository 1, is the root document, Doc B is a child of Doc A that lies in repository 2, and Doc C is a child of Doc B that lies in repository 1.

DFC does not report an error if you delete Doc B, but the containment object that bound Doc B to Doc A still exists in repository 1. As a result, if you log into repository 1, it looks to DFC as if Doc B is still part of Doc A. If you then perform operations on Doc A, DFC may report errors.

Cannot execute operations within transactions in multi-threaded environment

You cannot use classes of the `com.documentum.operations` package within transactions in a multi-threaded environment.

DFC does not support all OLE links

DFC supports OLE compound documents on Windows platforms. DFC supports only the following:

- OLE compound links in Word and PowerPoint
- External sheet and cell references in Excel formulas

In particular, DFC does not support OLE compound documents in Access, Outlook, or Visio. It does not support circular references in links.

DFC does not support mixing virtual and other compound documents

A compound document (for example, an OLE or XML document) cannot be a child in a virtual document.

Can't export an object with multiple formats in a single operation

If you add an object to an `IDfImportOperation` more than once, but with different formats, the operation exports the object to a single file on the file system.

Workaround: Use separate export operations for the different formats.

`IDfAttr.getDataType` and `IDfTypedObject.getAttrDataType` return unexpected values

The `getDataType` method of the `IDfAttr` interface and the `getAttrDataType` method of the `IDfTypedObject` interface can return unexpected values. The datatype they return for an

attribute that is part of a query collection is the type that Content Server returns. This may differ from the datatype that the IDfSysObject interface specifies for that attribute. The type returned may vary with the server's underlying relational database.

Workaround: Structure your application in such a way that you do not need to invoke these method.

IDfType.isSubTypeOf returns FALSE if not a direct subtype

The isSubTypeOf method of a type object returns false if its argument is not a direct supertype of the given type.

IDfTime.equals returns true if time difference is less than one second

The equals method of an IDfTime object returns a value of true if the method's argument represents a time that is within one second of the time that the object represents.

IDfPersistentObject.getChildRelatives ignores setting of child_value

The getChildRelatives method of the class that implements the IDfPersistentObject interface always returns the child_id of the dm_relation object, regardless of the value of child_label.

Importing Windows folder containing a shortcut does not import the file or folder the shortcut refers to

If you use an import operation to import a Windows file system folder and its contents into a repository, and if the folder contains a shortcut to a file or folder in another folder, DFC does not import the file or folder the shortcut points to.

Workaround: Import the files or folders that the shortcut points to.

Importing 'in progress' XML document fails if DDS validation is ON

If `<allow_work_in_progress>` is turned on in the XML application configuration file, and you try to import a document that is invalid or not well-formed, the import fails if `<dds_validation>` is also turned on, regardless of whether the data dictionary validation succeeds or fails.

Workaround: Do not use the work in progress feature with data dictionary validation. Documentum does not support this use.

XML application cannot process xml:base attributes

We do not support the `xml:base` attribute as specified by the XLINK language. You cannot import XML files containing this attribute. In particular, you cannot import packages that use the SCORM standard for web-based learning systems.

Transformations lose whitespace

DFC 5.3 uses a newer version of Xalan than previous DFC versions used. The new Xalan conforms more closely to the XSLT standard with regard to emitting whitespace. This may make the output of existing transformations appear on fewer lines than they did with the earlier Xalan.

Workaround: If you want line breaks, put explicit commands for them in your template.

Can't check out replica, then check in original

If you check out a replica of a document, the system checks the original document out as well. If you check in the replica, the system checks the original back in as well.

However, DFC does not support checking out a replica, then subsequently checking in the original directly.

Can't use two repositories with the same name

In an environment in which two Content Server instances control repositories that have the same name and project to different connection brokers, some DFC methods fail to execute correctly if they try to access both repositories.

File download dialog box does not appear during http transfer operations

The file download dialog box does not appear when attempting to perform file transfer operations such as checkout, view and edit in Webtop.

Workaround: This issue appears to be related to the security settings in Microsoft Internet Explorer 7. If the security settings in IE7 are set to medium (or higher), you will not be prompted to download the file. Adding the site as a trusted site, as well as customizing the security settings for trusted sites to allow prompting for file downloads, may resolve the issue.

To change the IE7 security settings:

1. In IE7, go to **Tools>Internet Options>Security** tab.
2. Select Trusted sites. Click the **Custom level** button in the section entitled **Security level for this zone**.
3. In the **Security Settings>Trusted Sites Zone** dialog box, scroll to the **Downloads** section and enable **Automatic prompting for file downloads**.
4. Click **OK** twice to exit the **Security Settings** and **Internet Options** dialog boxes.
5. Close and restart browser.

You should now be able to checkout and view files using Webtop in IE7.

Problems with importing a document on Macintosh platform

Importing a document on Macintosh platform does not import (and later export) the 'File Comments' of the document, viewable through the 'Get Info...' operation.

Replication of objects with aspects is not supported

This release of Content Server does not support replicating objects that have aspects.

Problems with netegrity multidocbase

The copy operation from one SSO to another SSO does not work.

Not possible to prevent renditions from deletion

It is not possible to lock a rendition to prevent it from being deleted. For example, when you try to export an object, and at the same time another user with appropriate permission over the object deletes it, the deletion succeeds and the you will get an error.

Issue when downgrading DFC

You cannot install CIS product (CIS installer also installs a DFC) when a more recent DFC has just been uninstalled in this machine.

Downgrading CIS:

If you plan to install a lower version of CIS after installing and uninstalling CIS version 6.5, you must also uninstall other components that were installed with CIS. If you don't uninstall these components, they will not be updated. The procedure below describes which components to uninstall and the required order.

To uninstall embedded components:

The procedure below assumes you already uninstalled DFC and the Add/Remove window of the Control Panel is still open. Uninstall the embedded components in the given order.

1. Select **Documentum Application Server** and click **Change/Remove**.
2. Select **Documentum Service Wrapper** and click **Change/Remove**.
3. Select **Documentum DFC Runtime Environment** and click **Change/Remove**.

You can now install a lower version of CIS.

Export operation of folder with special characters fails

The export operation of folder with special characters results in an exception.

Technical Notes

This section provides configuration and usability notes for current product features. No such information is available for this release of Documentum Foundation Classes.

Required configuration for machine-only application access control tokens

If you are using application access control tokens configured to be valid only when sent from applications on particular host machines, you must set the `dfc.machine.id` key in the `dfc.properties` file used by those client applications. The key must be set to the machine ID of the host from which the AAC token is sent.

Ensuring a correct `jaxp.properties` File

Some operating systems (for example, AIX) may install a `jaxp.properties` file, or that directory may contain a `jaxp.properties` file that was intentionally placed there. The purpose of the `jaxp.properties` file is to specify the way DFC implements the JAXP parser, by providing the class names for the JAXP factories. An empty `jaxp.properties` file may cause DFC to use the default implementation provided by the operating system.

A `jaxp.properties` file in the `JAVA_HOME/lib/` directory overrides the information in the manifest in `dfc.jar`, which points to the XML parser and transformers that come with DFC. This may lead to errors such as the following:

```
Content transfer: Server returned HTTP response code: 500 for URL:  
http://torcanary:8003/webtop/wdk5-xmlutil?operation=4
```

In the case of a WDK application, the application server's error log shows the following error:

```
javax.xml.parsers.FactoryConfigurationError:  
Provider null could not be instantiated:  
java.lang.NullPointerException
```

To resolve this problem, you can do one of the following:

- Rename the `jaxp.properties` file on your local system
- Add lines similar to the following to the `jaxp.properties` file:

```
javax.xml.transform.DfTransformerFactory=org.apache.xalan.  
processor.TransformerFactoryImpl  
com.documentum.xml.jaxp.DfSAXParserFactory=com.documentum.  
xerces_2_3_0.xerces.jaxp.SAXParserFactoryImpl  
com.documentum.xml.jaxp.DfDocumentBuilderFactory=com.  
documentum.xerces_2_3_0.xerces.jaxp.  
DocumentBuilderFactoryImpl  
javax.xml.parsers.DocumentBuilderFactory=com.documentum.  
xerces_2_3_0.xerces.jaxp.DocumentBuilderFactoryImpl  
javax.xml.parser.SAXParserFactory=com.documentum.xerces_2_3_0.  
xerces.jaxp.SAXParserFactoryImpl  
javax.xml.transform.TransformerFactory=org.apache.xalan.  
processor.TransformerFactoryImpl
```

Adjust the code for the specific Xerces version.

You can also edit the `jaxp.properties` file to use a different JAXP parser from the one packaged with DFC. For more information, refer to *XML Application Development Guide*.

Setting permissions for multiple DFC users on Unix systems

DFC uses an interprocess locking mechanism that uses lock files. This mechanism relies on allowing different OS users to write the lock file. The Unix operating system creates lock files using the file mode creation mask (`umask`) of the current process. Depending on the value of this mask, the operating system might assign different read/write permissions to the current user and to other users. In that case, an `IOException` can occur when a second user tries to acquire the lock. Two users doesn't have to be logged in at the same time for this problem to occur.

In order to avoid this problem, all users must be able to read and write the lock file. You can use the shell command `umask 011` in the application server's startup script to accomplish this.

Maximum number of results to retrieve per source by a query search

The maximum number of results to retrieve per source by a query search is set in the parameter `maxresults_per_source` of the `DFCfull.properties` as follows:

```
dfc.search.maxresults_per_source=350
```

Changing the `dfc.properties` dynamically on Windows

Documentum Server Manager provides the facility to change the `dfc.properties` dynamically on Windows. Properties definitions and values can be found in `dfcfull.properties`, and is located in `$DOCUMENTUM_SHARED/config`.

Copying a keystore from a machine to another does not work

Copying a keystore from a machine to another does not work by design.

WARNING: Keystores are not swappable, copyable or shareable. They are sensitive security information. If one needs, for any reason, to change a keystore, then stop the client, delete the keystore, and restart the client. On restart, the client will create a new keystore. All the privileges associated with the destroyed keystore, however, will be lost.

Documentation

This section describes the documentation related to this product.

Documentum Foundation Classes documentation set

These documents comprise the Documentum Foundation Classes documentation set for this version of the product:

- *Documentum Foundation Classes Release Notes*, version 6.5 (this document)
- *Documentum Foundation Classes Installation Guide*, version 6.5 (P/N 300-007-209)
- *Documentum Foundation Classes Development Guide*, version 6.5 (P/N 300-007-210)
- *DFC Online Reference (Javadocs)*, version 6.5 (P/N 300-007-211)

The following documents provide additional information to make using DFC easier:

- *Documentum System Migration Guide*, version 6.5 (P/N 300-007-225)

DFC 6.5 differs from earlier versions of DFC in significant ways. Refer to the migration guide if you are upgrading from an earlier version.

- *Content Server Fundamentals*, version 6.5 (P/N 300-007-197)

A conceptual description of the capabilities of Content Server and how to use them. The material in this manual is key to understanding most DFC interfaces.

- *XML Application Development Guide*, version 6 (P/N 300-007-228)

A description of the XML-related capabilities of Content Server, and an explanation of how to design applications that exploit those capabilities.

Obtaining the correct documentation

The Powerlink website (<http://Powerlink.EMC.com>) contains the downloadable packages for specific product versions along with the release notes, installation guide, and other relevant documentation associated with the product version. To locate product documentation, navigate to Support > Technical Documentation and Advisories > Software ~ D ~ Documentation, then select the product name and version number.

Note: The majority of Content Management products are listed under Software D > Documentum ?, where ? = a letter, or letters, in the alphabet.

Product documentation that is available online from the application (as online help) does not appear as a separate item; it is automatically downloaded and installed with the software.

Documentation corrections and clarifications

This section lists corrections to the product documentation.

Aspect deployment

Aspect modules are deployed using Documentum Composer. For more information, please see the *Documentum Composer User Guide*.

DMCL/DFC compatibility issues

The following are some known compatibility issues you may encounter when running version 5.3 custom applications with the Java DMCL emulator that ships with version 6.

String values no longer truncated by default

In DMCL/DFC 5.3 and before, if a value that is too large is set into a string attribute then the excessive data is silently discarded with no error. If you try setting a 37 byte value into a 32 byte attribute then the last 5 bytes are silently discarded.

This past behavior can be considered data corruption and is often dangerous because the user typically doesn't know that it happened.

DFC D6 now throws an exception if you try to overrun the size of an attribute.

In order to support backward compatibility DFC has a tunable preference to enable or disable the new behavior. The preference `dfc.compatibility.truncate_long_values` can be set to true to silently throw away data as in the past. The default for this preference is false. This default is chosen to avoid data loss (even though it is incompatible with previous versions).

When you encounter this new exception, the preferred solution is to carefully examine the application and resolve the real source of the problem. Chances are that silently discarding the data is not the application's desired result.

If fixing the application is not an option then you can set the preference in `dfc.properties` to enable truncation.

Routers no longer supported

Routers are no longer supported as of DFC version 6. Developers should now use workflows, instead.

Nameless getFile()

In DFC version 6, a `getFile` on a nameless file will return the content ID as the name of the file. In earlier versions, the object name was returned as the name of the file. The new behavior is a deliberate change in order to support object names that contain I18N characters.

Failed save() or checkin() operations

When a `save()` or `checkin()` operation fails on a `sysobject`, the object is automatically reverted and any unsaved changes to the object are lost. Clients should reapply any necessary changes before attempting to save or check in the object again. Earlier versions did not automatically revert objects when `save()` or `checkin()` operations failed.

Compatibility preferences

The following are preference settings that can help customizations from prior releases continue to run without incident.

- `dfc.compatibility.allow_weak_disconnect`

This setting is used to control whether a session can be disconnected through a weak session handle. By default, this is not allowed, because it is dangerous and often a source of application bugs. It is much safer to always disconnect through the strong handle that was obtained at "connect" time.

For compatibility with certain older applications, you can use this option to enable disconnect through a weak session handle. It is recommended that you avoid using this option. The application should be fixed instead. Enabling this option can allow application logic bugs to cause difficult to detect session management problems.
- `dfc.compatibility.allow_weak_setdocbasescope`

This setting is used to control whether a session can set repository scope through a weak session handle. By default, this is not allowed, because it is dangerous and often a source of application bugs. It is much safer to always setDocbaseScope through the strong handle that was obtained at "connect" time.

For compatibility with certain older applications, you can use this option to enable setDocbaseScope through a weak session handle. It is recommended that you avoid using this option. The application should be fixed instead. Enabling this option can allow application logic bugs to cause difficult to detect session management problems.
- `dfc.compatibility.truncate_long_values`

This setting is used to control the behavior when a value is too large to fit into the target attribute. When set to true, DFC will silently discard any extra data that will not fit in the target field. This is backward-compatible behavior with older DFC and DMCL versions, but is dangerous because it can result in data loss or corruption. When set to false, DFC will throw an exception when the value cannot fit in the target attribute.
- `dfc.compatibility.return_null_when_no_values`

This setting is used to control the behavior when a program asks for a value at index 0 from a repeating attribute that has no values. When set to true, DFC will return null when no values exist. This is backward-compatible behavior, but means that a failed request for index 0 behaves differently than a failedrequest for index 1. When set to false, DFC will throw an exception when no repeating values exist. Throwing an exception is a newer behavior that is more consistent with other repeating attribute accesses. It means a request for a non-existent entry at index 0 will return the same exception as a non-existent entry at index 1.

- `dfc.compatibility.preserve_session_info_messages`

This setting is used to control whether informational messages returned by the server at session establishment time are preserved. These messages are usually ignored by applications, and sometimes the messages actually get in the way if the application is not careful. It is typically more convenient to discard these unnecessary messages to avoid problems. If, however, your legacy application relies upon the informational messages, then you can set this preference to *true* so that they are preserved.

setString()

DFC version 6 is now more strict about what it accepts as valid input for the "attributeName" argument of the `setString` call.

In 5.3, DFC would accept arbitrary strings, which were passed directly on to DMCL. Sometimes, these strings included additional arguments or syntax that was specific to DMCL and not correct input for the DFC method. This was not a documented feature of the interface, but just happened to work, and some programmers took advantage of it.

For example, some older programs would pass the array index in the attribute name like this:

```
setString("keywords[0]", "value");
```

This is, strictly speaking, incorrect syntax. In 5.3 it just happened to work because of the way it was implemented. The method parameter name is "attributeName" and the string "[0]" is not part of the attribute name. It is the attribute index for a repeating attribute.

Repeating attributes are supposed to be accessed as follows:

```
setRepeatingString("keywords", 0, "value");
```

Another example is that of a date pattern. Sometimes, old programs would try to pass a date pattern by concatenating it with the attribute name this way:

```
setString("a_retention_date,mm/dd/yyyy", "12/17/1999");
```

This is also incorrect. The method parameter name is "attributeName" and the string ",mm/dd/yyyy" is not part of the attribute name. It is a date pattern. Date patterns are specified in DFC as follows:

```
setTime("a_retention_date", new DfTime("12/17/1999", "mm/dd/yyyy"))
```

DFC version 6 no longer accepts these incorrect values for the "attributeName" parameter of the "setString" call. Legacy code that was incorrectly using the "attributeName" parameter of "setString" should be updated to use the correct DFC APIs as described above.

setDocbaseScope() method deprecated

The `IDfSession.setDocbaseScope()` method has been deprecated. `setDocbaseScope()` makes a stateful change to the session. This makes the session unshareable. Unfortunately, Web applications have a tendency to share sessions frequently, and were not aware of this problem. If any code in a Webtop environment performed a `setDocbaseScope()` call, it could potentially cause problems for other threads in Webtop that were sharing the same session.

The primary purpose of `setDocbaseScope()` was to talk to another repository. A new method, `IDfSession.getRelatedSession`, has been introduced that should be used instead. This allows you to obtain a session to another repository in a much safer manner.

The old `setDocbaseScope()` method will still work in most situations, because of safety changes made in DFC version 6. In order to enforce this safety, however, there are restrictions. `setDocbaseScope()` can only be issued on a session that you obtained from a factory (`IDfSessionManager.getSession()` or `IDfSessionManager.newSession()`). If the session is obtained from a non-factory (like `IDfPersistentObject.getSession()`) then you cannot use `setDocbaseScope()` on that session.

If your existing application uses `setDocbaseScope()` and you get a `DFC_SESSION_HANDLE_RESTRICTION` exception, then you should either try updating your application, or, if that is not possible, you can set the following property in `dfc.properties`:

```
dfc.compatibility.allow_weak_disconnect=true
```

Session management enhancements

In DFC 5.3, session management problems were often obscure and hard to find. DFC version 6 has additional capabilities to help protect against and find session management bugs. Internally, DFC version 6 implements this additional protection through the use of session handles. Instead of returning a reference to the actual session, DFC returns a reference to a session handle. The session handle implements the `IDfSession` interface, so applications should not be able to tell the difference. Session handles help detect and protect against errors such as attempting to use sessions after they have been disconnected, or disconnecting someone else's session.

For most applications, developers should notice no difference. Some applications that have been using potentially risky session management practices might now throw an exception. Some exceptions can occur because the application has a bug. In this case you might see a message similar to:

```
DFC_SESSION_HANDLE_STALE
```

Other times, the exception might be thrown because the application is using a risky or dangerous session management pattern. In this case you might see an exception similar to:

DFC_SESSION_HANDLE_RESTRICTION

If one of the new exceptions is encountered, the best practice is to update the application to resolve the problem. Exceptions usually occur because the application is doing using bug prone session management techniques; it would be best to fix the application.

If fixing the application is not an option, there are compatibility options that can be enabled in `dfc.properties` to allow the older program to still work (in its potentially buggy state). Here are some of the compatibility options:

`dfc.compatibility.allow_weak_disconnect`

This setting is used to control whether a session can be disconnected through a weak session handle. By default, this is not allowed, because it is a potential source of application bugs. It is much safer to always disconnect through the strong handle that was obtained at "connect" time.

For compatibility with certain older applications, you can use this option to enable disconnect through a weak session handle. It is recommended that you avoid using this option. The application should be fixed instead. Enabling this option can allow application logic bugs to cause hard to detect session management problems.

`dfc.compatibility.allow_weak_setdocbasescope`

Used to control whether a session can set the repository scope through a weak session handle. By default, this is not allowed, because it is a potential source of application bugs. It is much safer to always `setDocbaseScope` through the strong handle that was obtained at "connect" time.

For compatibility with certain older applications, you can use this option to enable `setDocbaseScope` through a weak session handle. It is recommended that you avoid using this option. The application should be fixed instead. Enabling this option can allow application logic bugs to cause hard to detect session management problems.

DFC Development Guide: Documentation of do- method signatures

Page 147 of the DFC Development Guide states the following: The signatures for these methods are documented in Appendix.

The information does not appear in an Appendix, but in the section titled Signatures of Methods to Override on page 154 of the DFC Development Guide.

Software Media, Organization, and Files

This section describes the media in which the software is available, the organization of the product components in the available media, and the file names for all available product components which can be downloaded.

Software Media

This product is available as an FTP download from the Powerlink website (<http://Powerlink.EMC.com>). You should have received instructions through email regarding how to download products.

Organization

The Powerlink website (<http://Powerlink.EMC.com>) provides access to <https://emc.subscribenet.com/control/dctm/index>, which provides a complete listing of the products available for download.

Files

The following modules\files comprise the contents of this release:

- Documentum Foundation Classes (DFC) for Windows, Version 6.5 (exe)
- Documentum Foundation Classes (DFC) for Solaris, Version 6.5 (tar)
- Documentum Foundation Classes (DFC) for Linux, Version 6.5 (tar)
- Documentum Foundation Classes (DFC) for HP-UX, Version 6.5 (tar)
- Documentum Foundation Classes (DFC) for HP-UX IA64, Version 6.5 (tar)

- Documentum Foundation Classes (DFC) for AIX, Version 6.5 (tar)
- Documentum Foundation Classes (DFC) for *platform*, Version 6.5 (exe or tar)
- Documentum Foundation Classes (DFC) Release Notes, Version 6.5 (pdf)
- Documentum Foundation Classes (DFC) Development Guide, Version 6.5 (pdf)
- Documentum Foundation Classes (DFC) Installation Guide, Version 6.5 (pdf)
- Documentum Foundation Classes (DFC) Javadocs, Version 6.5 (zip)
- Documentum XML Application Development Guide, Version 6.5 (pdf)

Installation

The instructions for installing the product are provided in the *Documentum Foundation Classes Installation Guide*, version 6.5 (P/N 300-007-209). This section describes any additional installation issues.

Ensure that your system meets the requirements listed in [Chapter 4, Environment and System Requirements](#) before installing the software.

Troubleshooting and Getting Help

Documentum's technical support services are designed to make your deployment and management of Documentum products as effective as possible. The *Customer Guide to EMC Software Support Services* provides a thorough explanation of Documentum's support services and policies. You can download this document from the Powerlink website (<http://Powerlink.EMC.com>) by navigating to: Support > Support > Request Support > Software Customer Guide and Offerings.