

EMC® Documentum® System

Version 6.5

Upgrade and Migration Guide P/N 300-007-225 A01

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

Copyright © 1994 - 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.

Table of Contents

Preface	9
Chapter 1 Upgrade and Migration Overview	11
Upgrade and migration.....	11
Understanding migration.....	12
Order of new product installation.....	13
Order of system updates	13
Changing the database OS and version	15
Changing the content store location	15
Chapter 2 Planning System Size and Enhancing Performance	17
Planning the system size	17
Planning for performance.....	18
Common problems in Server performance	19
Common problems in web application performance	20
Chapter 3 Planning the System Migration	23
Changes in supported environments.....	23
Version 6.5 changes that impact Content Server upgrade or migration	26
Migrating objects to lightweight sysobjects (LWSOs).....	26
Changed behavior for attribute length	27
Maximum accepted string lengths in Documentum query language (DQL) statements	27
Required configuration for machine-only application access control tokens.....	27
Audit trail entries for dm_startedworkitem enhanced	28
DFC does not support linked store storage areas	28
External storage.....	28
DFC does not support optical storage devices.....	28
DFC Full format specifications no longer accepted	29
SYNC_REPLICA_RECORDS administration method	29
LDIF file changes.....	29
Obsolete dmcl.ini keys	29
New dfc.properties key to turn off trusted login	29
DQL changes.....	30
POSITION keyword no longer supported	30
CHANGE...OBJECT statement	30
DQL date literals enhancement.....	30
Behavior change after failed save or checkin.....	30
Trace method migration	31
New server.ini key	31
enable_workitem_mgmt obsolete	31
Mapping your current configuration.....	32

	Designing your version 6.5 configuration	36
	Addressing hardware concerns	36
	Upgrading third-party software.....	36
	Planning for global registries.....	36
	Mapping your version 6.5 configuration.....	37
	Planning upgrade and migration to version 6.5.....	38
	Setting up a test environment	38
	Client-first migration	38
Chapter 4	Interoperability and Compatibility	41
	Mixed Version Compatibility.....	41
	Guidelines for determining mixed version compatibilities.....	42
	Product-specific limitations	44
	Cross product dependencies.....	48
	Webtop, Digital Asset Manager	48
	Extended Search.....	48
Chapter 5	Migrating Content Server	49
	Migrating the Content Server	49
	Copying the global registry repository	50
	Rebuilding or upgrading fulltext indexes	51
	Configuring login tickets for backward compatibility.....	51
	Using DQL to migrate content to an XML Store	52
	Migrating custom Content Server methods.....	52
	Migrating DocApps and BOF2 modules.....	53
Chapter 6	Migrating WDK and Webtop Applications	55
	Overview	55
	Migrating WDK and Webtop applications.....	55
	6.0 changes that impact WDK and Webtop migration.....	57
	Architectural changes	57
	Presets	57
	Streamline deprecated	58
	<i>mainex</i> component is new default.....	58
	Themes deprecated.....	58
	Content transfer applet removed	59
	Implementation changes	59
	Right-click context menus	59
	Toolbar hidden by default	59
	Keyboard shortcuts (hotkeys).....	60
	Tab order configuration	60
	Auto complete.....	61
	Dynamic filters.....	61
	Configuration service modifications	61
	Invalid commands hidden by default.....	62
	Data grid enhancements.....	62
	Row selection	62
	Multi-object select.....	64
	Resizable columns	64
	Fixed column headers	65
	Miscellaneous enhancements.....	65
	Assigning relationships.....	65
	Lifecycle enhancements	65

	Drag and drop improvements	66
	Preference persistence	66
	6.5 changes that impact WDK and Webtop migration	66
	WDK	66
	Webtop	67
	Email conversion to EMCMF format	67
	Page refresh enhancements	67
	Multi-select drag and drop	68
	HTTP or UCF choice enhancement	68
	Deep export	68
	Modal pop-up dialogs	68
	UCF performance improvements in Webtop	69
	Content transfer dialog	69
Appendix A	Migrating DMCL APIs to DFC	71
	Overview	71
	Methods with no corresponding DFC method	71
	Methods with corresponding DFC methods	72
Appendix B	Object Type and Property Changes for version 6.5	83
	New object types	83
	Changed object types	85
	Deprecated or obsolete properties	91
	Properties added conditionally	91
	Deprecated or obsolete object types	93
	Changed properties	93
Appendix C	Configuration Settings in WDK-based Application Deployment	95
Appendix D	dfc.properties	99
	Overview	99
	Changes to existing key names	99
	dmcl.ini key migration to dfc.properties	102
	Obsolete dmcl.ini and session configuration options	104
	Obsolete dfc.properties keys	106

List of Figures

Figure 1.	System installation order, new Documentum system	13
Figure 2.	System update order, existing Documentum system	14

List of Tables

Table 1.	Upgrade and migration of product components	12
Table 2.	Browser environments not supported in 6.x	23
Table 3.	Application server environments not supported in 6.x	24
Table 4.	Databases not supported in 6.x	25
Table 5.	External products supported not supported in 6.x	25
Table 6.	Content Server and database server host worksheet	33
Table 7.	Application server host worksheet.....	34
Table 8.	Index server host worksheet	34
Table 9.	Client machine worksheet	35
Table 10.	Customized components worksheet.....	35
Table 11.	Mixed Version Configuration	43
Table 12.	Summary of product-specific limitations	44
Table 13.	Steps to migrate the Content Server from version 6 to version 6.5.....	49
Table 14.	Interaction between global versus local row selection settings	63
Table 15.	DMCL API methods and corresponding DFC methods	72
Table 16.	New Object Types.....	83
Table 17.	Changed Object Types	85
Table 18.	Deprecated and obsolete properties	91
Table 19.	Deprecated and obsolete computed properties	91
Table 20.	Properties added conditionally	92
Table 21.	Deprecated or Obsolete Object Types	93
Table 22.	Mandatory configuration before deployment	95
Table 23.	Optional configuration before deployment.....	95
Table 24.	Optional configuration after deployment	96
Table 25.	Name changes for existing dfc.properties for version 6.5	99
Table 26.	dfc.properties keys migrated from dmcl.ini file.....	103
Table 27.	Obsolete session configuration options	104
Table 28.	Obsolete dfc.properties keys.....	106

Preface

This guide focuses on the steps necessary to upgrade or move an existing EMC Documentum 5.3 or 6 implementation to the new EMC Documentum 6.5 platform. This guide does not focus on new features, except when a new feature changes or replaces existing behavior in custom applications.

Intended Audience

This guide is for EMC Documentum administrators who are tasked with upgrading or moving an existing EMC Documentum 5.3 or 6 implementation into the EMC Documentum 6.5 platform and developers who have created custom applications that need to move from EMC Documentum 5.3 or 6 to the EMC Documentum 6.5 platform.

Document scope

This guide shows you how to upgrade a Documentum system and migrate your customizations to the upgraded Content Server. To assist you in the upgrade process, refer to *Content Server Installation Guide* for more detailed planning information.

For migration of your customizations, this guide takes a version 5.3 or 6 implementation and has it work essentially the same way in version 6.5. You can take advantage of some but not all of the new features of version 6.5. This guide also shows you how to disable new behaviors where you do not want them and activate some features that have been deprecated or "turned off" by default for version 6.5. For information on how to implement new features in your custom client application, refer to the developer documentation for the product, for example, the *Web Development Kit Development Guide*.

The safe harbor releases for migration to version 6.5 are versions 6 and 5.3 SP6. If you are upgrading from a version earlier than 5.3, you must upgrade first to version 5.3, then to version 6.5.

Additional documentation

This guide provides overview and planning information. For details on specific procedures see these guides:

- *What's New in Documentum 6.5*
- *Content Server Installation Guide*
- *Content Server Administration Guide*
- *Documentum Foundation Classes Development Guide*
- *Web Development Kit and Webtop Deployment Guide*
- *Web Development Kit Development Guide*
- *Documentum Foundation Services Development Guide*
- *Documentum Foundation Services 6.5 Deployment Guide*

Revision History

The following changes have been made to this document.

Revision History

Revision Date	Description
July 2008	Publication for version 6.5

Upgrade and Migration Overview

This chapter covers in broad terms what is meant by upgrade and migration, what this guide covers, and where you might look for additional information.

These topics are included:

- [Upgrade and migration, page 11](#)
- [Understanding migration, page 12](#)
- [Order of new product installation, page 13](#)
- [Order of system updates, page 13](#)
- [Changing the database OS and version, page 15](#)
- [Changing the content store location, page 15](#)

Upgrade and migration

Upgrade is available for certain Documentum applications such as Content Server. Upgrade changes an existing installation to a new version. For information on upgrading Content Server, refer to *EMC Documentum Content Server Installation Guide*. You can install a new version 6.5 Content Server and migrate your existing repository to the new Server. For information on migrating your repository, refer to [Chapter 5, Migrating Content Server](#)

In migration, you move your customizations to a new instance. If you install a new Content Server, you need to migrate existing Server customizations such as DocApps or DAR files, and business objects. Some applications such as Webtop or the WDK framework require migration rather than upgrade. [Table 1, page 12](#) shows the components that need migration or upgrade.

Table 1. Upgrade and migration of product components

Component	Migrate	Upgrade
Content Server (5.3 or higher)	X	X
Custom DocApp (5.3 or higher)	X	
SBOs	X	
TBOs	X	
Java methods	X	X
Custom WDK/Webtop apps	X	

Note: DocApps, SBOs and TBOs (BOF2 version), and Java methods bundled as SBOs will continue to work in an upgraded Content Server. To edit them, you must create a Composer project. Refer to the "Managing Modules" chapter of *Documentum Composer User Guide*.

Check the installation or deployment guide for each application that you are upgrading to find instructions for upgrading or migrating applications to the new version. For information on migrating WDK-based customization, refer to [Chapter 6, Migrating WDK and Webtop Applications](#).

Before upgrade and migration, check the interoperability of all products and platforms in the system. Refer to [Chapter 4, Interoperability and Compatibility](#).

Understanding migration

Migrating from version 5.3 or 6 to version 6.5 is a straightforward process. Your task is to clearly document your current configuration, plan your version 6.5 configuration, then upgrade the individual system components in a sequence that will minimize impact on your users.

Migration can be separated into two basic tasks:

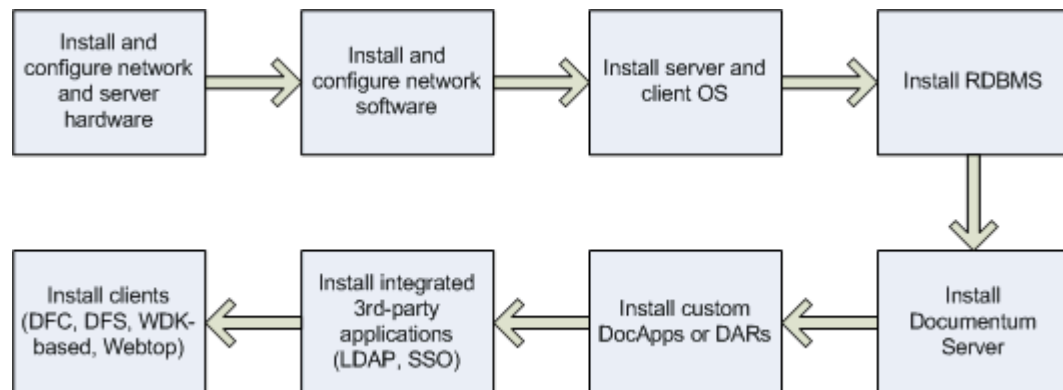
- Install and configure version 6.5 software.
- Move configurations and customizations to the new servers.
 - Make necessary changes to enable any features you want to keep.
 - Make necessary changes to disable any new features you do not want.
 - Make necessary changes to enable any new features for existing custom components.

Most of the new features of version 6.5 are enabled by default. For those feature that are not enabled by default, this guide explains the steps for enabling the new feature.

Order of new product installation

Figure 1, page 13 shows the order in which new Documentum system installations should be installed and deployed. The "server" in this diagram is the host for the RDBMS, Content Server, or Index Server.

Figure 1. System installation order, new Documentum system



Order of system updates

Upgrades to existing Documentum system installations may require uninstallation of some products that you are upgrading. Check the installation guide for recommendations. WDK-based clients version 6 such as Webtop or Web Publisher do not have an installer and thus cannot be upgraded. For those clients, you must migrate your existing customizations to a new WDK or Webtop runtime deployment.

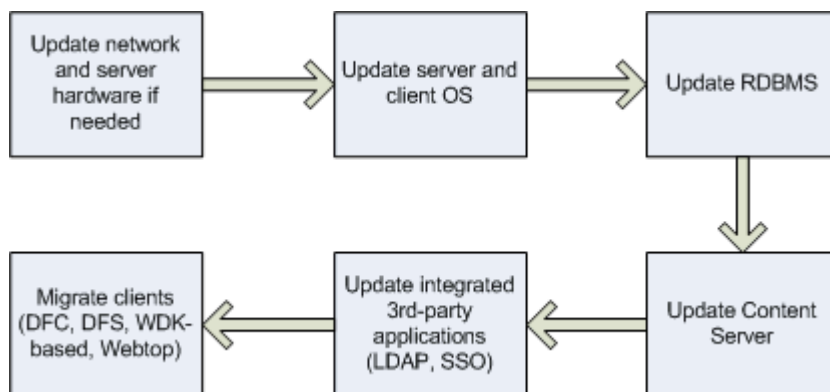
For upgrades that require uninstallation, perform the uninstall procedure in the reverse of the order shown for new installations. For example, if you need to upgrade the LDAP server OS, upgrade the Webtop application server and its OS, and then migrate your customizations to the new Webtop 6.5, proceeding in the following order:

1. Uninstall WDK 5.x. (Skip this step if you are migrating from 6.0 to 6.5).
2. Uninstall the WDK application server.
3. Update the LDAP OS.

4. Update the application server OS.
5. Update the application server software.
6. Deploy Webtop 6.5 on the new application server instance.
7. Migrate your WDK-based customizations to the Webtop 6.5 instance.

Figure 2, page 14 shows the order in which system components should be updated. The "server" in this diagram is the host for the RDBMS, Content Server, or Index Server.

Figure 2. System update order, existing Documentum system



Caution: For Content Server, host OS, or RDBMS upgrades, make sure the product version is supported by the Content Server version you are installing. For application server OS or server upgrades, make sure the product version is supported by the WDK-based application you are installing. This information is in the product release notes.

Chapter 4, *Interoperability and Compatibility* provides information on version compatibilities between interoperating Documentum products. When there are version compatibility restrictions between interoperating products, upgrading one product requires upgrading the interoperating product(s) to the same exact version or major version family. In most cases, version compatibility restrictions result from artifacts that a client product installs into the repository that leverage new functionality in Content Server. In these cases, upgrading the Content Server before the client application is especially important.

Changing the database OS and version

When migrating your database to a new OS (host) and database version, complete the migration first, before upgrading Content Server. Upon completion of the database migration, run the Content Server configuration program to re-establish the repository with the new database instance. Then upgrade the Content Server to upgrade the entire repository.

Refer to the database vendor documentation for information on migrating repository database files to a new database instance. The Content Server configuration program connects Content Server to the new database host unless the database connection string, database owner name, or password has changed.

Changing the content store location

You can move a content store to a new location. Refer to *Content Server Administration Guide* for information on moving file storage areas to a new location. The database location cannot be changed.

Planning System Size and Enhancing Performance

Upgrading a system presents an opportunity to change to new host environments. The following topics will help you plan your system size and improve performance:

- [Planning the system size, page 17](#)
- [Planning for performance, page 18](#)
- [Common problems in Server performance, page 19](#)

Planning the system size

Use the system sizing spreadsheet available on Powerlink to assist you in planning your system. On the [Powerlink site](#) Products menu, click Software D > Documentum 6 Platform > System-Sizing Spreadsheet.

The system sizing spreadsheet takes into account your estimated first-year and subsequent years input of common types of documents (Word, PPT, PDF, HTML, XML, images, MPEG), number of light and heavy users, and number of custom types. These figures are modified by WDK and BPM configurations. The output is the estimated number of CPUs and memory required for Content Server, Index agent and server, WDK application server, and RDBMS server. Additional calculations are performed for a document transformation server and BPS server if selected.

Hardware and network planning are essential for system scalability. For example, a load rate of 125,000 images per hour during an 8-hour day could lead to a billion objects in less than 5 years. In a regulated environment, those object may need to be online. The following considerations are recommended from a billion-object benchmark study with 1000 concurrent users:

- Set query resource limits in production. If a query uses large table scans, then its response time will be in terms of hours rather than minutes given the size of the tables.

- Pre-test every application query on a reasonably-sized database prior to production.
- Eliminate poorly formed queries. Poorly formed queries in a small application might significantly degrade when mixed with a larger repository.
- The full-text index server should allocate three times more storage to the full-text index than to the content filestores.

To determine the system load expectations

1. Determine number and locations of functional users, including end users, administrators, and developers.
2. Determine average and peak level data traffic (frequency and size of content transfers) for users.
3. Determine rate of ingestion of new content and magnitude of content to be migrated.
4. Determine indexing requirements for new and migrated content.

Planning for performance

There are many factors that affect performance of a Documentum system. These factors include:

- Number of users
- Frequency and duration of user sessions
- Size of files transferred by users
- Network capacity
- Storage and RAM capacity on hosts for RDBMS, Content Server, Index, application server or other

These factors are used by the System Sizing spreadsheet to help you calculate your size needs. On the [Powerlink site](#) Products menu, click Software D > Documentum 6 Platform > System-Sizing Spreadsheet.

In Documentum 5 systems, DMCL tracing was used to analyze performance. In Documentum 6, you create a DFC trace. There are 34 tracing parameters in `dfc.properties`. Logging and tracing is fully described in the Content Server Administration Guide. Some sample scripts to convert a trace file to Microsoft Excel and to analyze repetitive calls are available on the [EMC Developer Network](#).

You can monitor application server memory usage using graphical display tools for Java garbage collection statistics such as Samurai or the IBM Diagnostic Tool for Garbage Collector. Monitor the frequency of full garbage collection to determine whether the JVM heap size needs to be adjusted. You can generate a Java heap dump for out of memory exceptions and analyze them with the Sun Heap Analyzer Tool (HAT), the IBM Heap Dump Analyzer, or the Your Kit Java Profiler. Classes that consume memory to check in

a heap dump are `com.documentum.web.failover.AttributeWrapper`, `FormHistory`, and `NavigationObservable`.

Common problems in Server performance

The most common problems in Content Server performance include the following:

- Application design and customization
 - Chatty or redundant application code, the most common cause of failure (40% in 1995 IEEE study)
 - Complex object models
 - Poor memory and cache management
- Network
 - High latency due to physical or logical limitations
 - Overburdened shared network
- Undersized Server resources (inadequate memory and CPU size)
- Unmanaged or underused RDBMS resources
 - RDBMS not regularly monitored and tuned
 - Performance and caching features not used
- Unrealistic expectations (did not use realistic benchmarks)

A highly available infrastructure must be carefully designed to fit your business requirements. There is no single solution that fits all.

The following solutions are available to help with these problems.

High availability Documentum Server clusters — Server clusters (also called Server sets) can be active-active or active-passive. In an active-active cluster, there are two active load-balanced web application servers, two active sets consisting of a Content Server and connection broker,, one active RDBMS with clustered standby server, one primary database with one synchronized standby, and one primary content store with one synchronize standby. In an active-passive cluster, everything is the same except that there is only one active Server plus connection broker set, with another set as standby.

These cluster configurations provide partial high availability coverage with increased scalability. The clusters can be managed with Documentum Administrator.

Redundant connection brokers — Connection brokers (formerly known as docbrokers) can be configured to automatically reroute users to Content Servers that are online. Connection brokers can load balance user connections across multiple Content Servers

using identical proximity values for connection brokers. Refer to *Content Server Administration Guide* .

Replication — Replication can be configured either as read/write or read-only.

Disaster recovery — Disaster recovery is not the same as high availability. It assumes a total loss of the production data center. Disaster recovery servers are separate and independent from the main center. They share no resources, and each has an independent network infrastructure for WAN replication. Both systems have the capacity to carry out full normal and emergency workloads. They must be maintained to be completely compatible.

Failover for disaster recovery is manual, not automatic. Clients will be affected.

Enhancing query performance — The Content Server Administration Guide describes how to monitor query performance using the Update Statistics administration tool. It also describes how to limit poorly-performing subqueries for users who belong to a large number of groups.

Common problems in web application performance

Performance guidelines for WDK-based applications are published in the *Web Development Kit Development Guide*. These include the following areas:

- Configuration
 - Allocate sufficient Java memory
 - Limit the number of allowable HTTP sessions
 - Set a lower default page size
 - Limit browser history
 - Turn on value assistance caching
 - Configure response compression and caching static elements
- Customization
 - Implementing actions
 - Creating objects
 - Limiting cookie lookup

Search performance — Several search performance guidelines are available in this same guide.

Content transfer performance — The following factors may increase content transfer performance:

- Limit the number of imports per user transaction in the importcontainer configuration
- Increase UCF session timeout, for example, from 250 to 500 seconds, in WEB-INF/classes/ucf.server.config.xml.
- On Windows clients, turn off virus scanning for archives.
- On Windows clients, turning off virus scanning for the Documentum/ucf subdirectory of the user's home directory, for example, Documents and Settings\my_name\Documentum\ucf.
- On Windows clients, turning off virus scanning for the Java executable directory and subfolders.

Planning the System Migration

Migrating a system requires planning. You have to know your starting point, choose a destination, then pick the best route to get there. This chapter provides some practical advice for plotting your course from version 6 to version 6.5 and later versions. Topics in this chapter include:

- [Changes in supported environments, page 23](#)
- [Version 6.5 changes that impact Content Server upgrade or migration, page 26](#)
- [Mapping your current configuration, page 32](#)
- [Designing your version 6.5 configuration, page 36](#)
- [Planning upgrade and migration to version 6.5, page 38](#)

Changes in supported environments

[Table 2, page 23](#) shows changes in supported browser environments between versions 5.3.x and 6.x. Not every item in the table applies to every product. Refer to the release notes for each product for detailed information on supported software environments.

Table 2. Browser environments not supported in 6.x

Browser version	OS
Firefox 1.x	All
Firefox 2.0.x	Solaris 8, Solaris 9 Red Hat Linux 3 SuSE Linux Enterprise 9 Mac OS X 10.3.9 Novell Linux Desktop 9
IE 6 SP1	Windows 2000
Mozilla 1.7.x	All

Browser version	OS
Netscape 7.2	All
Safari 1.3.x	All

Note: The Microsoft browser VM and Apple Java 1.3.1 browser VM are not supported in Documentum 6.x.

Table 3, page 24 shows versions of application server environments that are not supported in 6.x:

Table 3. Application server environments not supported in 6.x

Application server version	OS
Apache Tomcat 5.0.28	All
Apache Tomcat 5.5.25 and 6.0.x	Windows 2000 Solaris 8, Solaris 9 AIX 5.2 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9 HP-UX 11
BEA WebLogic Server 8.1	All
BEA WebLogic Server 9.2 and 10	Windows 2000 Solaris 8, Solaris 9 AIX 5.2 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9
IBM WebSphere AS 5.1.x	All
IBM WebSphere AS 6.x	Windows 2000 Solaris 8, Solaris 9 AIX 5.2 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9
Oracle 10g (9.0.x)	All
Oracle 10g (10.1.x)	Windows 2000 Solaris 8, Solaris 9 AIX 5.2 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9 HP-UX 11

Application server version	OS
Sun Java System AS 7	All
Sun Java System AS 8	Solaris 8, Solaris 9

Note: Microsoft Internet Information Server (IIS) 5.0 is not supported as an external web server.

Table 4, page 25 shows database environment versions that are not supported in 6.x.

Table 4. Databases not supported in 6.x

Database version	OS
DB2 8.x	All
DB 2 9.1	Windows 2000 Solaris 8, Solaris 9 AIX 5.2
Oracle 9i Release 2 (9.2.x)	All
Oracle 10g Release 1 (10.1.x)	All
Oracle 10g Release 2 (10.2.x)	Windows 2000 Solaris 8, Solaris 9 AIX 5.2 HP-UX 11 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9
Sybase ASE 12.5.x	All
Sybase ASE 15.0.2	Windows 2000 Solaris 8, Solaris 9 Red Hat Enterprise Linux 3.0 SuSE Linux Enterprise 9
SQL Server 2000	All
SQL Server 2005	Windows 2000

Table 5, page 25 shows the unsupported third-party product versions for 6.x:

Table 5. External products supported not supported in 6.x

Category	Product
Operating Environment	Citrix MetaFrame Presentation Server 3.0
Operating Environment	Citrix MetaFrame Presentation Server 4.0

Category	Product
Operating Environment	eTrust Siteminder Policy Server 5.5
Operating Environment	RSA Access Manager 5.5
Operating Environment	VMWare GSX Server
Directory Server	Microsoft Windows 2000 Server Active Directory
Directory Server	Oracle Internet Directory 9.2.x
Other	Adobe Acrobat 5.0.x
Other	Adobe Acrobat 6.0.x
Other	Centera SDK 3.0
Other	Microsoft Office 2000

Version 6.5 changes that impact Content Server upgrade or migration

This section describes miscellaneous changes that may impact the migration to version 6.5.

Migrating objects to lightweight sysobjects (LWSOs)

Lightweight sysobjects (LWSOs) are useful if you have a large number of attribute values that are identical for a group of objects. This redundant information can be shared among the LWSOs from a single copy of the shared parent object. For example, Enterprise A-Plus Financial Services receives many payment checks each day. They record the images of the checks and store the payment information in sysobjects. They will retain this information for several years and then get rid of it. For their purposes, all objects created on the same day can use a single ACL, retention information, creation date, version, and other attributes. That information is held by the shared parent object. The LWSO has information about the specific transaction.

The administrative method, `MIGRATE_TO_LITE`, migrates objects to LWSOs. You specify the type to turn into a lightweight type and the shareable parent type. One use case is to split a standard type up so that some attributes are in the parent type and the rest are in the lightweight type. Another case is to make the entire standard type into the lightweight type and create a brand-new type as the shareable parent type. After the method executes, each LWSO has its own private parent, so the parents are not

shared. At this point, you can reparent the LWSOs to shared parents and delete the now "orphaned" parents. For full information on migrating objects to LWSOs, refer to *Documentum High Volume Server Developers Guide*.

Changed behavior for attribute length

In DMCL/DFC 6 and before, if a value that is too large is set into a string attribute, the excessive data is silently discarded with no error. For example, if you try setting a 37-byte value into a 32-byte attribute, 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 6.5 now throws an exception if you try to overrun the size of an attribute. To support backward compatibility DFC has a tunable preference to enable or disable the new behavior.

You can set the preference `dfc.compatibility.truncate_long_values` 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.

If you prefer to use the pre-6.5 behavior, set the following `dfc` preference in `dfc.properties` to T:

```
dfc.compatibility.truncate_long_values
```

When you get 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 your desired result. If fixing the application is not an option, you can set the preference in `dfc.properties` to allow truncation.

Maximum accepted string lengths in Documentum query language (DQL) statements

The maximum length of a character string literal in a DQL statement is now governed by the maximum allowed by the underlying relational database. In previous releases the DQL parser for some databases enforced a smaller maximum.

Required configuration for machine-only application access control tokens

If you are using application access control (AAC) tokens configured to be valid only when sent from applications on particular host machines, you need to set the

dfc.machine.id key in the dfc.properties file used by those client applications. The key needs to be sent to the machine ID of the host from which the AAC token is sent.

Audit trail entries for dm_startedworkitem enhanced

For dm_startedworkitem events, the string_4 property of the audit trail object now records the performer of the work item.

DFC does not support linked store storage areas

In version 6.5, DFC will not support linked store storage areas. As a consequence, the following items are deprecated:

- The dm_linkedstore object type, which represents linked store storage areas
- The dmi_linkrecord object type, which records the links between a linked storage area and file stores
- The CLEAN_LINKS administration method, which removes orphaned link records if needed

External storage

If you are using an external storage area and the plugin is configured to execute on the client host, you need to reconfigure the plugin to execute on the server. In version 6.5, DFC does not support executing the plugin on the client. To configure the plugin to execute on the server, set the a_exec_mode property of the storage object to F (FALSE). The storage object is one of dm_extern_file, dm_extern_free, or dm_extern_url, depending on the type of external storage you are using.

DFC does not support optical storage devices

DFC does not support optical storage devices with version 6.5.

DFC Full format specifications no longer accepted

The DFC methods, such as *setFile*, that previously accepted a full format specification no longer do so. As of version 6.5, those methods accept only a format name, such as *txt* or *word*, for the format argument.

SYNC_REPLICA_RECORDS administration method

The SYNC_REPLICA_RECORDS administration method is obsolete. It was used to complete the migration from DocPage Server 3.x to eContent Server 4.x. Neither of these server versions is currently supported. References to this method have been removed from the documentation.

LDIF file changes

Use of "true", "false", "1" or "0" as values for Boolean properties in the LDIF file is deprecated. Also, the list of accepted properties in the file has been updated.

Obsolete dmcl.ini keys

Two keys added in 5.3 SP4, `max_file_size` and `max_backup_index`, are obsolete in the 6.5 release.

New dfc.properties key to turn off trusted login

By default, applications running on the Content Server host are allowed to make repository connections as the installation owner without presenting a password. This is called a trusted login. If an application, such as Documentum Administrator, that has an explicit login dialog box is installed on a Content Server host, a user is able to login as the installation owner without a password using a trusted login.

This release introduces a new `dfc.properties` key to turn off trusted logins if you do not want to allow trusted logins through such applications. The key is:

```
dfc.session.allow_trusted_login
```

Setting this key to false requires users to always provide a password, even when logging in as the installation owner.

DQL changes

The following are changes to the DQL.

POSITION keyword no longer supported

The POSITION keyword, previously supported in SELECT queries against the fulltext index, is no longer supported.

CHANGE...OBJECT statement

Previously, using the CHANGE...OBJECT statement was restricted to custom object types. With this release, the statement may be used to change any type so long as the remaining restrictions as listed in the DQL Reference manual description of CHANGE...OBJECT are not violated.

DQL date literals enhancement

You can now specify 'utc' in a date literal in a DQL statement. The new syntax for date literals is:

```
DATE('date_value[utc]' [, 'pattern'])
```

You can define *date_value* using any of the valid character string formats representing a date, or it can be one of the keywords that represent dates.

If utc is included, Content Server assumes that the specified *date_value* is UTC time. The specification of utc is not case sensitive.

Behavior change after failed save or checkin

In version 6.5, if a save or checkin of an object fails, DFC will revert the object automatically before returning the object to the application issuing the save or checkin. Reverting the object removes any changes made to the object before the attempted save or checkin. The application must then reapply any changes made prior to the save or checkin before reattempting the save or checkin operation. This behavior is different from how failed saves or checkins were handled by the prior DMCL. The DMCL in prior releases simply marked the object as in an error state and returned it to the application.

Trace method migration

In version 6.5, the tracing implementation for client-side tracing is changed and enhanced. The trace method is replaced with the `IDfSession.setServerTraceLevel` method. For complete details, refer to the *Content Server Administration Guide*.

The Trace API method in existing scripts will continue to work, but its implementation is changed. In version 6.5, the level 0 turns tracing off and any nonzero value turns tracing on. If a file name is specified on the Trace method command line, the trace information is recorded in that file. If no file is specified, the trace information is placed in the file specified in the `dfc.properties` file, in accordance with the new tracing implementation.

New server.ini key

Version 6.5 introduces a new `server.ini` key for Windows platforms. The key allows you to configure the maximum size of the listener queue for Content Server connection requests.

Content Server creates a socket listener for incoming connection requests with a maximum backlog set to 200 by default. On Windows platforms, you can reset that maximum if needed. To do so, set the `listener_queue_length` key in the `server.ini` file. Set the key to a positive integer value. Content Server passes the specified value to the Windows Sockets call `listen()`.

enable_workitem_mgmt obsolete

The `enable_workitem_mgmt` key controls whether permissions to perform certain workflow actions are enforced. The affected actions are:

- Acquiring a work item
- Delegating a work item
- Halting and resuming a running activity
- Changing a work item's priority

If the key is set to T (TRUE), any user can perform those actions. The key is F (FALSE) by default.

Mapping your current configuration

The following system configuration diagrams and sample worksheets provide a starting point for documenting the infrastructure of your current system. You might already have similar diagrams from which you can get much of this information. If you do not, be sure to keep a copy of your version 6 plan to help with future migrations. Take the time to verify that any existing diagrams reflect the current configuration.

Complete one copy of the table below for each server host and client configuration used in your current system, for example, Content Server, fulltext indexing server, ECI server, application server.

Table 6. Content Server and database server host worksheet

Item	Value
Hardware and Processors	
Memory	
Operating system and version	
Content Server version	
RDBMS and version	
Repository size	Number of objects: Storage space required:
Global Registry?	<input type="checkbox"/> Yes <input type="checkbox"/> No
Java/JRE version	
DFC version	
Other product version	
Other product version	
Other product version	

Table 7. Application server host worksheet

Item	Value
Hardware and processors	
Memory	
Operating system and version	
HTTP Server version	
Java version	
DFC version	
Other product and version	
Other product and version	
Other product and version	

Table 8. Index server host worksheet

Item	Value
Hardware and processors	
Memory	
Operating system and version	
HTTP server version	
Java version	
DFC version	
Other product and version	
Other product and version	
Other product and version	

Table 9. Client machine worksheet

Item	Value
Operating system and version	
Browser and version	
Java version	
Other product and version	
Other product and version	
Other product and version	

Table 10. Customized components worksheet

Product	Customized Components	Customization type	Customization Description	Disposition
				<ul style="list-style-type: none"> • 6.5 Compatible • Needs changes • Obsolete
				<ul style="list-style-type: none"> • 6.5 Compatible • Needs changes • Obsolete
				<ul style="list-style-type: none"> • 6.5 Compatible • Needs changes • Obsolete

Designing your version 6.5 configuration

This section discusses some of the design decisions you need to make before implementing your version 6.5 configuration. *Departmental* systems are configurations where the Content Server, RDBMS, and global registry all reside on the same host machine. *Enterprise* systems are configurations containing multiple Content Servers, data repositories, and distributed services to improve performance in high traffic or geographically disbursed environments.

Addressing hardware concerns

Verify that the hardware you are currently using will continue to meet your needs for the foreseeable future. In particular, if you have been hosting more than one server on a single machine, for example, Content Server and an application server, this is a good time to divide the functions between two or more server hosts to boost performance.

Upgrading third-party software

Prior to migrating to version 6.5, verify that the third-party software you are currently using with version 6 is still supported, and upgrade to supported versions as necessary. See the *What's New in Version 6.5* document for a list of supported software.

Planning for global registries

You need to designate one of the repositories in your version 6.5 system as the *global registry*. The global registry is a central location used to store common objects used by all repositories, such as SBO (server-based objects) locations, BOCS (Branch Office Caching Service) settings, and user settings. You need to decide which of your repositories will be enabled as the global registry.

If you already have a version 5.3 or 6 global registry, you can use it with a repository that you upgrade to 6.5. Refer to for compatible versions of Content Server, global registry, and client.

During repository configuration, you are prompted to choose one of the following options:

- Use the current repository as a global registry

You need to provide a user login name and password for the global registry user in the repository you are currently configuring. Record the login name and password; you will use this login name and password to configure other repositories in your system to allow them to access the global registry. The local DFC instance is also configured to access this global registry.

- Specify a different repository as the global registry

You need to provide the repository name and the login credentials (user login name and password) of the global registry user in that repository. The DFC instance on the current host is configured to access the remote global registry repository.

- Do later

If you choose this option, you can delete the `dfc.bof.registry.repository`, `dfc.bof.registry.username`, and `dfc.bof.registry.password` from the `dfc.properties` file and rerun the DFC installer on this host in order to designate the global registry repository at a later time. Version 6.5 requires a global registry, so clients should not connect to the system until the global registry is configured.

Regardless of whether you designate the repository as a global registry, the global registry user is created all repositories. The global registry user, who has the user name of `dm_bof_registry`, is the repository user whose account is used by DFC clients to connect to the repository to access required service-based objects and user information. The user has read access to objects in the `/System/Modules` only.

- If you configure the repository as a global registry, you provide the user login name and password for the user and the user state is set to Active.

This can be any arbitrary user login name and password. Do not use the repository owner's credentials or the installation owner's credentials.

- If you do not configure the repository as a global registry, the user is created with a default value for the login name and the user state is set to Inactive.

If you later enable the repository as a global registry, use Documentum Administrator to change the user state to Active and provide the user with a user login name and password that you choose. Refer to the *Content Server Installation Guide* for instructions on enabling the repository as a global registry.

Mapping your version 6.5 configuration

For each server host and client configuration, complete a planning document. You can use the same forms used for mapping your current configuration (see [Mapping your current configuration](#), page 32).

Planning upgrade and migration to version 6.5

Now that you know your starting point and your destination, you can choose the best upgrade and migration path. The recommended configuration is a homogeneous version 6.5 system. The migration paths described below allow your applications to continue working and minimize impact on your users, but your users will not get the full benefits of version 6.5 features until the migration is complete.

Setting up a test environment

Before migrating your production system, EMC Documentum recommends that you set up a test environment that includes the same hardware, RDBMS, and software configurations as your production system, including a copy of your production repository. This allows you to practice migrating your systems, as well as troubleshoot any migration problems before committing changes to your production system.

Client-first migration

If your system uses only Webtop, DFS, custom DFC, or custom WDK clients, you have the option of migrating the client applications first. Refer to the installation or deployment guide for the client application for detailed instructions.

ACS and BOCS version compatibility and migration — DMS requires BOCS version 6.5 in order to request content.

An encrypted store requires BOCS version 6.5.

Parallel streaming from ACS will be used only if both ACS and UCF (WDK or DFS applications) are version 6.5. Parallel streaming from BOCS will be used only if ACS, BOCS, and UCF are version 6.5.

Note: When you upgrade the BOCS to version 6.5, you must update the BOCS version specification in the global registry using Documentum Administrator. For BOCS 6.0, specify the version as 2.0. For BOCS 6.5, specify the version as 2.1.

WDK clients compatibility and migration — WDK-based clients are compatible with version 5.3 and version 6 Content Server. Some WDK version 6.5 features will be available, such as those that run in JavaScript on the client. Other features will not be available until you complete the migration to version 6.5, such as lightweight sysobjects, data partitioning, batch processing and scoping..

These are the steps to migrate from version 5.3 or 6 to version 6.5, migrating the clients first.

1. Upgrade the application server and client browsers.
2. Enable the global registry in a version 6 repository in order to support version 6 client features that require a global registry. Refer to *Documentum Content Server Installation Guide* for instructions. Version 6.5 clients with a version 5.3 global registry is not supported.
3. Upgrade the Content Servers in place.
4. Configure one Content Server as the version 6.5 global registry. If you had a global registry in version 6, you can upgrade that server in place using the same settings.

Interoperability and Compatibility

Unless otherwise noted, any 5.3 SP2 (or greater SP) or 6.x SPx product can coexist with any other 5.3 SP2 (or greater SP) or 6.x SPx product on the same host. Products that *only* coexist on the same host do not interoperate with each other. In addition, unless otherwise noted, all 6.x SPx products can interoperate with all other 6.x products. Interoperability takes place when different client applications perform operations on the same object instance in a repository.

[Mixed Version Compatibility, page 41](#), provides detailed information for determining compatibility between different version client applications and Content Server. The *Cross-product dependencies and interoperability* section of each product's release notes provides a list of products depended on by a product and additional products with which a product generally interoperates.

Mixed Version Compatibility

Under most conditions, 5.3 SP2 (or greater SP) or 6.x SPx clients can work with Content Servers that are from a different major version family, 6.x SPx or 5.3 SP2 (or greater SP), respectively. This kind of system configuration is referred to as a mixed version configuration. You might want to set up a mixed version configuration when you want to migrate only the client or server side of your production environment to 6.5. In a mixed version configuration, most customers migrate the server side of their production environment first.

Note: You can also have a mixed version configuration between different products in the same version family (6.0, 6.0 SP1, and 6.5, for example).

In a mixed version configuration, functionality provided by the higher version product is generally not available to the lower version product. The mixed version products work fine together, but the new functionality provided by the higher version product is not exposed or it is disabled. In some cases (see [Guidelines for determining mixed version compatibilities, page 42](#) and [Product-specific limitations, page 44](#)), new functionality provided by the higher version product may cause a lower version product not to work properly.

Guidelines for determining mixed version compatibilities

The client application major version family (for any 5.3 SP2 or later version) must match the major version family of the global registry (5.3 SP2 or later) in order to work against any 5.3 SP2 (or greater SP) or 6.x SPx Content Server. For example, 5.3 SP2 (or greater SP) client applications require a 5.3 SP2 (or greater SP) global registry to work against a 5.3 SP2 (or greater SP) or 6.x SPx Content Server. Similarly, 6.x SPx client applications require a 6.x SPx global registry to work against a 5.3 SP2 (or greater SP) or 6.x SPx Content Server.

Client applications also contain a combination of artifacts (modules, aspects, Type-Based Objects (TBOs), or Service-Based Objects (SBOs)) that contribute to mixed version compatibility or incompatibility, depending on how these artifacts were developed. [Table 12, page 44](#) lists mixed version compatibilities and limitations between Documentum client products and Content Server for the 5.3 SP2 (and greater SP) and 6.x SPx major version families.

In a typical deployment, the application server contains the modules, aspects, and TBOs for a client application while the global registry contains the SBOs. The application server also contains a version of Documentum Foundation Classes (DFC), typically installed by the client application, that is the same version as the client application.

For 5.3 SP2 (or greater SP) client applications, always compile the modules, aspects, and TBOs using the same JDK version (1.4) used by the Documentum Foundation Classes (DFC) on the application server. Compiling these artifacts with a higher version JDK (1.5, for example) is not supported because object binaries compiled with JDK 1.5 are not guaranteed to run on JDK 1.4.

For 6.x SPx clients applications, best practice is to compile 6.x SPx artifacts with JDK 1.5 because there might be JDK 1.5 specific calls in the artifact code preventing the artifact from successfully compiling with JDK 1.4. However, if a module, aspect, or TBO can be successfully compiled using JDK 1.4, then it can run against DFC 6.x SPx on an application server running JDK 1.5. For SBOs, it is recommended to always use JDK 1.4 for 5.3 SP2 (or greater SP) client applications and JDK 1.5 for 6.x SPx client applications.

[Table 11, page 43](#) summarizes which combinations of client application, DFC and JDK versions on the client application server, global registry version, and JDK versions of the modules, aspects, TBOs and SBOs enable access to Content Server 5.3 SP2 (or greater SP) or 6.x SPx.

Table 11. Mixed Version Configuration

Client Application Version	Application Server Environment	JDK Modules/ Aspects/ TBOs Compiled With	Global Registry Version	JDK SBOs Compiled With	Content Server Version
5.3 SP2 (or greater SP)	DFC 5.3 SP2 (or greater SP)	None JDK 1.4	5.3 SP2 (or greater SP)	None JDK 1.4	5.3 SP2 (or greater SP), 6.x SPx
	JDK 1.4	JDK 1.5	5.3 SP2 (or greater SP)	None JDK 1.4	No valid Content Server versions
6.x SPx	DFC 6.x SPx JDK 1.5	None JDK 1.4, JDK 1.5	6.x SPx	None JDK 1.5	5.3 SP2 (or greater SP), 6.x SPx
Note: None indicates that the client application does not have any modules, aspects, TBOs, or SBOs.					

Product-specific limitations

If a product is not included in this section, then it can be deployed in a mixed version configuration as described in [Table 11, page 43](#).

Based on the guidelines in [Guidelines for determining mixed version compatibilities, page 42](#), [Table 12, page 44](#) summarizes whether a specific client product version works with a 5.3 SP2 (or greater SP) repository and a 6.x SPx repository. Unless otherwise noted, 5.3 SP2 (or greater SP) indicates any of the 5.3 service packs, SP2 through SP6, and 6.x SPx indicates 6.0, 6.0 SP1, or 6.5.

Note: Many products consist of multiple installed components (WAR file, DAR file or DocApp, for example). Mixing versions of these components (6.0 WAR file and 6.5 DAR file, for example) for a particular (6.5) product version is not supported.

Table 12. Summary of product-specific limitations

Product	Product Version	Supported Content Server Versions	Unsupported Content Server Versions	Notes and Exceptions
Archiving Services for SAP	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
Business Activity Monitor	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	Note 1
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
Content Intelligence Services	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	Note 5
Content Services for SAP	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
Content Transformation Services	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
Digital Asset Manager	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	

Product	Product Version	Supported Content Server Versions	Unsupported Content Server Versions	Notes and Exceptions
Documentum Administrator	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	
	6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
Documentum Collaborative Services	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	Note 3
Documentum Compliance Manager	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP), 6.x SPx	None	Note 2
	6.5	6.0 SP1 or 6.5	5.3 SP2 (or greater SP), 6.0	
Document Image Services	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	Note 1
File Share Services (FSS)	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
Forms Builder	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	Note 1
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
FTP Services	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
Physical Records Manager	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	Note 2
Process Builder	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	Note 1
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	

Product	Product Version	Supported Content Server Versions	Unsupported Content Server Versions	Notes and Exceptions
Process Engine	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	Note 1
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
Process Integrator	6.5	6.x SPx	5.3 SP2 (or greater SP)	Note 1
Process Services for SAP	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP), 6.x SPx	None	
	6.0, 6.0 SP1	5.3 SP2 (or greater SP), 6.x SPx	None	
	6.5	5.3 SP5, 6.x SPx	5.3 SPx (except SP5)	
Records Manager	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP), 6.x SPx	None	Note 2
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
Retention Policy Services	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP), 6.x SPx	None	Note 2
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
Site Caching Services	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	
TaskSpace	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	Note 1
Web Publisher	5.3 SP2 (or greater SP)	5.3 SP2 (or greater SP)	6.x SPx	
	6.x SPx	6.x SPx	5.3 SP2 (or greater SP)	

Product	Product Version	Supported Content Server Versions	Unsupported Content Server Versions	Notes and Exceptions
Webtop	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	Note 4
WebDav Services	5.3 SP2 (or greater SP), 6.x SPx	5.3 SP2 (or greater SP), 6.x SPx	None	
<p>Note 1: Business Activity Monitor, Document Image Services, Forms Builder, Process Builder, Process Engine, Process Integrator, and TaskSpace must be the same exact version (6.5, for example) as each other and the Content Server, when all are using the same repository.</p> <p>Note 2: Physical Records Manager, Records Manager, and Retention Policy Services must be the exact same version (6.5, for example) as each other when all are using the same repository. Similarly, any other client (Webtop, for example) that interoperates with objects created by these products, must be at the same version.</p> <p>Note 3: Documentum Collaborative Services (DCS) 6.x SPx requires manual installation of a DAR file in order for Content Server to provide full support for new features in the 6.x SPx DCS client. Once this DAR file is installed, only 6.x SPx versions of the DCS client (and other clients) are supported by the 6.x SPx Content Server.</p> <p>Note 4: Queue management is not supported when using Webtop 5.3 SP2 (or greater SP) clients against a 6.x SPx Content Servers in which the 6.x SPx BPM TBO has been installed.</p> <p>Note 5: CIS 5.3 SP2 (or greater SP) and 6.0 require the same exact version of Documentum Administrator. CIS 6.x SPx requires Documentum Administrator 6.x SPx.</p>				

Cross product dependencies

Note: EMC recommends that you use DARs instead of DocApps (whenever possible).

Webtop, Digital Asset Manager

The WDK-based application WAR file contains scripts to upgrade a 5.3 SP2 (or greater SP) repository for subscriptions. Run the DQL script `subscriptionInstall.dql` that is located under the root web application directory, in `webcomponent/install`. Taxonomy Manager support scripts are located in the directory `webcomponent/install/admin/tm`.

Extended Search

5.3 SP2 (or greater SP) cannot access Extended Search functionality in a 6.5 content server. ,the artifact can , so you should always compile these artifacts with JDK 1.5. For 6.x SPx client applications, the modules, aspects, and TBOs can be compiled using the JDK version 1.4 or 1.5, because DFC 6.x SPx is compiled with JDK 1.5 and JDK 1.5 is backward compatible with JDK 1.4.

Migrating Content Server

If you are installing a new instance of Content Server 6.5 and migrating data from a previous version on a separate host, you need to follow a procedure somewhat different from an upgrade.

This chapter addresses any variation from the basic scenario to known issues surrounding the configuration of your version 6.5 server.

- [Migrating the Content Server, page 49](#)
- [Copying the global registry repository, page 50](#)
- [Rebuilding or upgrading fulltext indexes, page 51](#)
- [Configuring login tickets for backward compatibility, page 51](#)
- [Using DQL to migrate content to an XML Store, page 52](#)
- [Migrating custom Content Server methods, page 52](#)
- [Migrating DocApps and BOF2 modules, page 53](#)

Migrating the Content Server

Migrating Content Server version 5.3 or 6 to 6.5 occurs in three phases:

- Back up your existing data.
- Run the version 6.5 installer.
- Configure the new Content Server to use your existing repository.

These are the recommended steps for preparing and migrating your version 6 Content Server to version 6.5.

Table 13. Steps to migrate the Content Server from version 6 to version 6.5

Step	Documentation
1. Back up your repository.	Several third-party tools are available for backup.

2. Clean up your repository.	<i>Content Server Administrator's Guide</i> , Content Repositories chapter, "Cleaning up repositories."
4. Run the Consistency Checker utility.	<i>Content Server Administration Guide</i> , Tools and Tracing chapter, "Consistency Checker."
5. Fix any errors identified by the Consistency Checker.	
6. Back up your cleaned, consistent repository.	
7. Install Content Server.	See <i>Content Server Installation Guide</i> .
8. Configure Content Server to use your existing repository.	See <i>Content Server Installation Guide</i> .

Copying the global registry repository

You can upgrade the global registry using one of the following methods:

- You can upgrade the current global registry repository to the Content Server version 6.5 global registry repository.
- You can create a new repository as the version 6.5 global registry repository and use the global registry copy utility to copy the objects from the current global registry repository to the new version 6.5 repository, then decommission the old global registry.

This is useful if you have multiple repositories and you do not want to upgrade all of them to version 6.5 at once. The older version repositories would continue to use the older version global registry and the upgraded repositories would use the new version 6.5 global registry.

To copy the global registry repository with the copy utility:

1. Run the Content Server configuration program to create a version 6.5 repository.
2. From the command line, run the repository copy utility to copy the global registry objects:

```
dmbasic -f globalReg6.5_copy.ebs -e EntryPoint - globalReg6.5 6.5User
6.5Pass destDB destUser destPass tempFileName
```

- globalReg6.5 is the name of the version 6.5 global registry repository from which to copy objects.
- 6.5User is the login name to use for the version 6.5 global registry repository.

- 6.5Pass is the password to use for the version 6.5 global registry repository.
- destDB is the name of the destination repository to which to copy the global registry objects.
- destUser is the login name to use for the destination repository.
- destPass is the password to use for the destination repository.
- tempFileName is the name of a temporary file used for internal processing. The directory for this file needs to exist, but not the file itself. You need to provide a full path name, but the file name does not require an extension.

You need to run this utility from the same machine on which the Content Server for the version 6.5 repository is running. Run this utility only once to the destination repository to avoid creating duplicate global registry objects in the destination repository.

3. Change each dfc.property to point to the new version 6.5 global registry repository as each DFC application is upgraded to version 6.5.
4. When all DFC applications have been upgraded to version 6.5, delete the version 6.5 global registry repository and upgrade the original version 6 global registry repository to version 6.5.
5. When the upgrade is completed, point the repository to the new global registry.

Rebuilding or upgrading fulltext indexes

If you are migrating from version 5.2.5 to version 5.3, then to version 6.5, you need to rebuild the fulltext index only once, when you move to 5.3. You do not have to rebuild fulltext indexes when you subsequently migrate from 5.3 to version 6.5.

If you are upgrading from 5.3 or 5.3 SP1, you need to rebuild your fulltext indexes.

If you are using fulltext indexing, are upgrading from 5.3 SP2 or SP3 and have applied the Get Well 4.3.1 hot fix, or you are upgrading from 5.3 SP4 or later, you do not need to rebuild your fulltext indexes to migrate to version 6.5.

Refer to the *Content Server Administrator's Guide* for fulltext rebuilding and upgrade procedures.

Configuring login tickets for backward compatibility

In an environment that includes mixed versions of Content Server, the login tickets generated by a particular Content Server might not be accepted by a Content Server at

another version level. A version 6.5 Content Server can accept a login ticket generated from any Content Server, regardless of the server's version level. A pre-5.3 SP4 Content Server cannot automatically accept login tickets generated by a Content Server at version 5.3 SP4 or higher. Also, a 5.3 SP4 or 5.3 SP5 Content Server cannot automatically accept login tickets generated by a version 6.5 Content Server.

When you are upgrading a production environment with multiple repositories or multiple Content Servers for one repository, there will likely be intervals when the Content Servers are at differing levels. To ensure that login tickets generated by a Content Server are backward compatible, set the `server_login_ticket_version` key in the `server.ini` file:

- Set the key to 1 to generate login tickets acceptable to a pre-5.3 SP4 Content Server
This setting is only valid on versions 6.5, 5.3 SP4, and 5.3 SP5 Content Servers.
- Set the key to 2 to generate login tickets acceptable to a 5.3 SP4 or 5.3 SP5 Content Server
This setting is only valid on version 6.5 Content Servers.
- Set the key to 3 to generate login tickets acceptable to version 6.5 Content Servers.
This setting is only valid on version 6.5 Content Servers.

The key defaults to 3 if not set.

Using DQL to migrate content to an XML Store

You can migrate XML files from an existing Documentum file store to an XML Store, between XML Stores, and out of an XML Store using an update DQL query. To migrate:

- Run DQL Query as `UPDATE dm_sysobject OBJECTS set a_storage_type = 'xhive_store_01' where a_storage_type = 'filestore_01' and a_content_type = 'xml'`

Note: This procedure migrates only the current version of the object.

Migrating custom Content Server methods

After Content Server upgrade, you must run the configurator tool to configure the internal Java method server. The configurator will write the location of your Java methods to the internal method server. The location of the methods directory is written to the file `web.xml` in the method server deployment directory, for example, `C:\Documentum\jboss4.2.0\server\DctmServer_MethodServer\deploy\ServerApps.ear\DmMethods.war\WEB-INF:`

```
<init-param>  
  <param-name>methodlocation-1</param-name>
```

```
<param-value>C:\Documentum\dba\java_methods</param-value>
</init-param>
```

Your custom Content Server methods located in %DOCUMENTUM%\dba\java_methods (Windows) or \$DOCUMENTUM/dba/java_methods (UNIX and Linux) will continue to work. If you are migrating to a new Content Server installation, copy the methods from this directory to the same folder location in the new Content Server installation.

Migrating DocApps and BOF2 modules

BOF version 2 modules and DocApps do not need to be changed when you upgrade a Content Server 5.3.x or 6.0 to 6.5. If you want to make changes to a DocApp or module on an upgraded 6.5 Content Server, create a project in Composer and add your BOF2 modules or DocApp. Refer to *Documentum Composer User Guide* for full instructions on working with modules and Documentum Archive (DAR) files.

Use the Composer project migration utility to migrate a DocApp or a DocApp archive to a DAR file: **New > Project > Documentum Project > Documentum Project from Repository DocApp**. Composer will generate a DAR file that can be installed in a new instance of Content Server or edited in place in an upgraded Content Server instance. Complete instructions are in the "Migrating DocApps" chapter of the *Composer User Guide*.

If you want your version 6.5 BOF2 modules to be used by DFC 5.3 clients, you need to:

1. Compile them for a Java 1.4.x target `<javac target=1.4>` to make them compatible with older virtual machines.
2. Compile them against DFC 5.3 rather than DFC 6.5 to ensure they do not accidentally reference new interfaces.

To migrate custom Business Objects in an environment of 5.3 SP6 clients that access a version 6.x Content Server, do the following:

- SBO

Install your 5.3 DocApps in the 5.3 SP6 global registry. Do not upgrade this global registry.

- Module or TBO

Make sure your code will work with DFC 5.3 SP6. It must compile with JDK 1.4.2 and must not use any classes or methods that are new in DFC 6.x.

Migrating WDK and Webtop Applications

These topics are included:

- [Overview, page 55](#)
- [Migrating WDK and Webtop applications, page 55](#)
- [6.0 changes that impact WDK and Webtop migration, page 57](#)
- [6.5 changes that impact WDK and Webtop migration, page 66](#)

Overview

EMC recommends that you install and configure WDK-Webtop 6.5 on an application server different from your existing WDK-Webtop installation and then migrate any customizations to the 6.5 WDK-Webtop 6.5 installation. Do not migrate any of your existing customizations that new 6.5 functionality can replace. To enable full 6.5 functionality, you will also need to upgrade the DocApps/DARs in the repositories (including global registries) that your WDK application accesses. For more information, see [Chapter 5, Migrating Content Server](#).

Migrating WDK and Webtop applications

To migrate WDK and Webtop applications from version 5.3 or 6.0 to 6.5:

1. Back up your customizations, if you have made changes in any of the following:

- Web application customizations:
 - `APP_SERVER_ROOT/webtop/custom` directory
 - Compiled custom classes in `APP_SERVER_ROOT/webtop/WEB-INF/classes` directory
 - Custom tag libraries in the `APP_SERVER_ROOT/webtop/WEB-INF/tlds` directory

See *Web Development Kit and Webtop Deployment Guide*.

- Application server startup file

Note: Do *not* migrate settings that the WDK installer added to your application server startup file.

See *Web Development Kit and Webtop Deployment Guide*.

2. EMC recommends but does not require that you uninstall DFC 5.x and any application that uses DFC 5.x and then reboot before deploying an application based on WDK 6.x. For uninstall procedures, refer to the 5.x product documentation—specifically, the *Web Development Kit and Webtop Deployment Guide*.
3. If necessary, update your application server software.
Refer to the supported application servers in the release notes for the WDK-based product.
4. Make the required setup changes to your Webtop WAR file, then deploy the Webtop WAR file.
See *Web Development Kit and Webtop Deployment Guide*.
5. Choose the features you want to use.
6. Copy the contents of your previous `/custom` directory, to the `/custom` directory on your new server. Copy custom Java classes and TLDs to the `WEB-INF` folder on your new server.
7. Recompile your custom 5.3 classes to ensure that the custom code still works and copy them and TLDs to the `WEB-INF` folder on your new server.
8. Enable, disable, and migrate your chosen features.
See *Web Development Kit and Webtop Deployment Guide*.
9. Test and fix your web application.
10. Deploy your web application to your production application server.
See *Web Development Kit and Webtop Deployment Guide*.

6.0 changes that impact WDK and Webtop migration

This section describes changes that may impact your migration of WDK/Webtop applications from version 5.3.x to version 6.0.

Architectural changes

This section describes changes to the overall architecture of WDK/Webtop that may impact migration.

Presets

Application presets provide a common framework for defining, populating, and accessing context-sensitive application configuration data.

Application configuration data encompasses user preferences, component configuration, and scope-specific component configuration. Application configuration data is open-ended; the characteristics of an application that can be configured are defined by an application developer. Examples of application configuration data include the following:

- what actions are available to a user
- what repository types users can create in a particular folder
- what work flow users in a particular role can start
- what columns a developer has made available to the user by default
- what columns a particular user has selected for display in the current view

The primary differentiating factor between presets and XML configuration files is that while XML configuration is performed by a developer prior to deployment, presets can be updated by users after the system is deployed. Webtop provides a GUI preset editor that can be used by business users to create presets for groups of users. Presets have precedence over XML configuration settings.

Presets can be used for many of the UI enhancements that were performed using XML configuration files in prior releases. Before migrating your customizations, look into what can be accomplished using presets, and consider implementing your customizations as presets. Presets offer significant benefits in terms of flexibility and maintainability.

For information on using presets, see the *Webtop User's Guide*.

Streamline deprecated

The Streamline interface was designed to provide quick access to the most commonly used commands. Version 6.5 introduces the right-click context menu, which provides the same ease of use in a way that is familiar to most users. The Streamline interface is now obsolete, and is disabled by default. However, it still ships with our product for backward compatibility. If you would like to enable the Streamline interface (assuming you've consulted your users, and they agree that they prefer the Streamline interface), use one of these methods:

- To re-enable Streamline view for all users and all HTTP sessions, add the following setting in `/custom/app.xml`:

```
<streamlineviewvisible>true</streamlineviewvisible>
```
- To re-enable Streamline view for one HTTP session, launch the main component with the parameter `entryPage` set to *streamline*.

For example, Webtop can be launched using a URL similar to

```
http://localhost:8080/webtop/component/main?entryPage=streamline
```

Once the Streamline view is enabled, it cannot be disabled again within the same HTTP session.

mainex component is new default

The *mainex* component is launched from `index.html` so that the JSP page defaults to the classic component. In your applications, you can either update your customizations to extend *mainex* or, if you choose to continue to use Streamline, call your custom component from `index.html`.

Themes deprecated

Version 6.5 introduces a new version of the *documentum* and *high contrast* themes. These themes enhance performance and maintainability, and provide additional "look and feel" benefits. Any 5.3 theme can be enabled in `app.xml`, but features that are new in 6.50 will not display a 6.0 theme or a custom theme that extends a 5.3 theme. If you want to use a custom theme that extends the *documentum* or *high contrast* theme, you will need to adjust your JSP pages to match the changes introduced in the JSP page design.

Content transfer applet removed

The content transfer applet is no longer supported. We recommend that you upgrade your custom applications to use the UCF file transfer utility.

Implementation changes

The following are changes to the implementation of controls in WDK/Webtop applications. Most of these features are enabled by default: the descriptions tell how to disable the new behavior if you prefer not to use it.

Right-click context menus

Webtop version 6 provides access to commonly used commands via a right-click context menu. This functionality provides an interface that is more familiar to users than the Streamline interface, which is now deprecated.

Context menus work in data grids configured for row selection (that is, rows that do not use check boxes). Context menus are defined in the <menugroup> element of an action configuration file. <actionmenuitem> identifies an action that can be performed on the object type that is specified in the scope of the configuration file. Context menus can be disabled along with row selection and other new data grid enhancements by setting the value of <desktopui>.<data grid>.<richui> to false. For more information on modifying and creating context menus, see the *Web Development Kit Development Guide*.

Fixed menus are also moving to XML. This has the benefit of allowing you to scope and extend XML-based menu definitions. JSP-defined menus are still supported for backward compatibility. For step-by-step instructions for migrating your JSP-based menus to XML, see the *Web Development Kit Development Guide*.

Toolbar hidden by default

The toolbar is no longer necessary with the introduction of the right-click menu, and so has been hidden by default. It is still present in the webtop\classic\classic.jsp JSP page, but its frame height is set to 0. You can display the toolbar by changing the line

```
String strRows = "0,*";
```

to

```
String strRows = "22,*";
```

Keyboard shortcuts (hotkeys)

Hotkeys provide a new mechanism for users to activate commands using the keyboard. Hotkeys are supported for the following controls:

- button
- link
- menuitem
- actionbutton
- actionlink
- actionmenuitem
- actionimage

Custom controls that extend these control tags inherit hotkey support if they add the appropriate attributes to the tag library. For complete information on implementing your own hotkeys, see the *Web Development Kit Development Guide*.

Hotkeys are enabled by default. To disable hotkeys for your entire application, modify the <hotkeys> element of the app.xml file, setting the <enabled> element to *false*.

Tab order configuration

WDK now supports the HTML attribute *tabindex* to determine the order in which controls receive focus when the user types the tab key. The following WDK controls support tab ordering in version 6.5:

- Text (also XFormsText, FileBrowse)
- Password
- TextArea (also XFormsTextArea)
- Button
- Link
- ActionButton
- ActionLink
- ActionImage
- Tree
- CheckBox
- Radio
- Image
- DropDownList (also DataDropDownList, ListBox, DataListBox)
- DateInput

- DateTime

Tab ordering is enabled by default. Tab ordering can be turned on or off for different environments in app.xml. We recommend that you turn off tab ordering in portal applications.

For full information on modifying tab order and adding tab order support to your custom components, see the *Web Development Kit Development Guide*.

Auto complete

The auto complete feature provides value assistance for drop-down list and text controls. The values are read from the data dictionary. Completion lists are stored by *autocompleteid* for each user. The values are not specific to an object type or repository.

Auto complete is enabled by default. The autocompleteid setting defaults to the form name concatenated with the control name. Auto complete can be disabled for the entire application by setting the <auto_complete>.<enabled> attribute to false in the app.xml file.

You can add auto complete functionality to any control that extends the WDK text or dropdown list controls. You can also create custom controls that do not extend text or dropdown list but can support auto complete. For full information, see the *Web Development Kit Development Guide*.

Dynamic filters

A new control, <dmf:datacolumnbeginswith>, allows you to filter the selection in data grids and dropdown lists. The filter is bound to the containing control on the JSP that implements IDataboundControl. The filter can be combined with the auto complete feature for added convenience. For full details on implementing dynamic filters, see the *Web Development Kit Development Guide*.

Configuration service modifications

In earlier versions of WDK, XML configuration elements had to be overridden in their entirety. This made applications "brittle" and difficult to maintain during upgrade. Each customized application essentially required its own WDK stack.

As a first step toward a single WDK stack for all applications, we have introduced the ability to merge XML into an existing configuration file. These scoped elements modify rather than override target elements in configuration files. The following modifications

can be made on the referenced configuration definition: insert, insertbefore, insertafter, replace, remove.

For example, rather than override an entire browsertree element and its sub-elements, you can use the merge syntax to reconfigure the items in the browsertree.

```
<scope role ="CSI_Investigator">

<component modifies="browsertree:webtop/config/browsertree.xml">

<insertbefore path="nodes.docbasenodes.node[componentid=homecabinet_classic]>
    <node componentid="investigations">
        :
        :
    </insertbefore>
<remove path="nodes.docbasenodes.node[componentid=administration] " />
</component>
</scope>
```

For full details on configuration service extensions, see the *Web Development Kit and Client Applications Development Guide*

Invalid commands hidden by default

A new setting in /webtop/wdk/app.xml, `<display>.<showinvalidactions>`, is an application-wide override to the `showifinvalid` setting on action controls. Setting this element to *false* (default) will hide actions that are not valid in the user's current application context, regardless of the setting in the `showifinvalid` attribute of the action control.

Data grid enhancements

Several substantial improvements have been made to the look and feel of data grid controls.

Row selection

Users can now click rows to select objects rather than a check boxes at the left margin. This feature is enabled by default, and requires no migration effort for data grids that leverage the `ActionMultiSelect` and `ActionMultiSelectCheckbox` controls to drive row/object selection. Defining a double click event for the row requires a minor JSP modification, the addition of a tag to the page that identifies the event/action and the arguments to invoke on double-click.

Data grids that use generic check boxes controls will require some migration effort. The enhanced data grid fires a set of client-side events for data grid initialization and selection changes: the developer can implement listeners for these events.

WCL/Webtop Locators (Choosers), which currently use generic check boxes controls to drive row selection, will have these client-side listeners written and registered with the data grid control as shipped and, apart from marking the generic check boxes on customized JSP pages, no additional work will be required for customized locators.

To disable row selection, add the following element to /custom/app.xml:

```
<desktopui>
  <datagrid>
    <richui>false</richui>
  </datagrid>
</desktopui>
```

The following table describes the interaction between the global row selection flag in app.xml and the data grid attribute.

Table 14. Interaction between global versus local row selection settings

app.xml <rowselection>	dmf:datagrid richui = " "	Result
false	true or false	Check boxes rendered, no mouse/keyboard row selection or context menus on any data grid
true	true	No check boxes rendered, row selection and context menus enabled for the data grid
true	attribute not specified (for example, migrated customizations)	No check boxes rendered, row selection and context menus enabled for the data grid
true	false	Check boxes rendered, no mouse/keyboard row selection or context menus on current data grid

For more information on implementing row selection and double-click handling, see the *Web Development Kit Development Guide*.

Multi-object select

Users can shift-click to select contiguous rows or ctrl-click to create a selection of rows not adjacent to each other. Users can also use the ctrl-a shortcut to select all items in the data grid, arrow keys to change the selection, and shift-arrow to select multiple contiguous items. ACLs, properties, and lifecycles can now be applied to several items at a time using multi-object selection.

Resizable columns

The Column Resizing feature requires additional markup within the data grid layout definition. In version 5.3, the data grid had no inherent notion of which of the rows (TR) inside its markup was the "header" row. Additionally, it had no idea which data field is "bound" to a cell (TD/TH) inside each row. These two concepts need to be marked using server side controls in order to:

1. Attach event handlers to respond to mouse interaction for column resizing.
2. Render persisted widths for each cell representing a data field, since this is how field widths will be persisted – per data field.

All WCL/Webtop Classic View Listing Components will contain this markup change as shipped. Existing customizations of those components with custom layout pages need to be modified to meet these new markup requirements in order to enable the feature.

On the JSP, the component author needs to add *resizable="true"* for headers of columns that are supposed to be resizable.

```
<dmf:datagridTh resizable="true">
. . .
</dmf:datagridTh>
```

If the listing component extends the DocList listing component, this is all that needs to be done. However, if the component is not a descendent of DocList, the following code needs to be added to the implementation of the component class:

```
public void onInit(ArgumentList args) {
    . . .
    initColumnWidths();
}
/**
 * Initialize column width preferences.
 */
protected void initColumnWidths() {
    m_widthsHelper =
        new DatagridColumnWidthPreferenceHelper
            (this, <preference id string>, CONTROL_GRID);
}

private DatagridColumnWidthPreferenceHelper m_widthsHelper;
```


Fixed column headers

Headings now stay in place while the user scrolls through the items in the data grid. To enable this behavior, custom components will need to implement header rows using the new element *datagridTh*. For more information, see the *Web Development Kit and Client Applications Development Guide*.

To globally disable fixed column headers, add the following element to */custom/app.xml*:

```
<desktopui>
  <datagrid>
    <fixedheaders>false</fixedheaders>
  </datagrid>
</desktopui>
```

Miscellaneous enhancements

The following enhancements do not have specific impact on migration beyond the fact that most of them can be disabled by setting a value in the *app.xml* file. It is important to evaluate these new features as they apply to any customizations you have made to your own applications. For more information on these features, see the *Webtop User's Guide*.

Assigning relationships

A user can create a relationship between any two documents in the repository. A user can also delete a relationship between two documents. This feature requires a version 6.5 Content Server and repository.

Lifecycle enhancements

Lifecycle is now displayed on the Properties Screen (part of the default attribute set). Lifecycle is exposed as a column in object lists. Users are able to apply a lifecycle on documents at creation time (and Import, Create, Checkin), or later, via the Properties screen. The process of applying a lifecycle to a document has been enhanced to include the ability to specify the initial lifecycle state and the alias set. The current success/error messages for the lifecycle-related actions displayed on the Message Bar that involve a lifecycle state change (Apply, Detach, Promote, Demote, Suspend and Resume) will be enriched to contain pertinent information about the object lifecycle state (previous and current).

Drag and drop improvements

When a user drags a document into a directory with an item of the same name, a popup menu displays to allow the user to choose between creating a new rendition of the repository file, or replacing the content of the existing file with the dropped file.

Preference persistence

User preferences are now stored in the global registry in addition to local cookies. The cookie size has been streamlined for scalability. This enhances reliability and makes the user's customized settings available from any client machine.

6.5 changes that impact WDK and Webtop migration

See the *Web Development Kit Development Guide* for more information about implementing these new features and the *What's New in Documentum 6.5* for summary information about these new features. For a list of controls, actions, and components that are new or have been removed, see the *Web Development Kit Reference Guide*.

WDK

LwSOs store metadata, common to many different (but similar) child objects, at the parent object level. This dramatically reduces the storage footprint and ingestion processing required for the child objects.

The actions and components that implement these features have changed. Compare your custom components to the corresponding 6.5 components to determine whether you need to merge your custom component with the corresponding 6.5 components associated with these features:

- Browsing
- Simple Search
- Advanced Search
- Attributes
- Properties

Webtop

The following new items apply to Webtop for Documentum 6.5:

- [Email conversion to EMCMF format, page 67](#)
- [Page refresh enhancements, page 67](#)
- [Multi-select drag and drop, page 68](#)
- [HTTP or UCF choice enhancement, page 68](#)
- [Deep export, page 68](#)
- [Modal pop-up dialogs, page 68](#)
- [UCF performance improvements in Webtop, page 69](#)
- [Content transfer dialog, page 69](#)

Email conversion to EMCMF format

Email message (.msg) files are imported as a type or subtype of the dm_message_archive type and are converted to the EMCMF format. The conversion of native email messages to EMCMF format is handled by UCF. By default, email conversion to EMCMF format is disabled. You can enable this feature in the app.xml file.

New actions and components have been created to implement these features. Compare your custom components to the corresponding 6.5 components to determine whether you need to merge your custom component with the corresponding 6.5 components for these functions:

- Import (for example, emfimport is a new component)
- Export (for example, export (type dm_message_archive) is a new action)
- View Properties and Listing pages (for example, the attributes (type dm_message_archive) is a new component)
- Search (for example, the search (type dm_message_archive) and advsearch (type dm_message_archive) are new components)

Page refresh enhancements

Webtop improves performance by reducing the amount of refreshes and by making better utilization of the AJAX framework. For example, a user chooses a folder in the browser tree to view a list of content contained in the folder. Before Documentum 6.5, there was first a refresh of the browser tree applet and then there was a refresh of the content list. In Documentum 6.5, there is no refresh of the browser tree. The browsertree (version

6.0) (Webtop) component is a new component and `JumpToBrowserTreeLocationAction` (Webtop) is a new action are related to this feature.

Multi-select drag and drop

Users can select multiple files and perform a drag and drop. The multi-select drag and drop functionality is available for all areas where single file drag and drop was previously available. For example, users can multi-select files and drag and drop them to another location in the repository. Multi-select drag and drop also works when exporting and importing multiple files to and from the local file system.

HTTP or UCF choice enhancement

Webtop 6.5 enables administrators to specify HTTP or UCF content transfer for different users within the same Webtop installation. Before Documentum 6.5, all users within the same Webtop installation had to use either HTTP or UCF content transfer. This feature is enabled and disabled in `app.xml`.

Deep export

Webtop provides the ability to export one or many folders and allow the structure of those folders to remain intact depending on the permission set of the files and folders. By default, deep export is disabled, and you have to enable it in `app.xml`.

Modal pop-up dialogs

This feature provides modal pop-up dialogs for action screens involving dialogs. A modal pop-up dialog is a child window which requires the user to interact with it before they can return to the parent application. This feature enhances performance and allows the user to see the context from where the action was launched. Previously, the user choose an action, the screen refreshed and took the user to a new screen. With modal pop-up dialogs, a new window pops up on top of the previous screen. The previous screen is viewable, but no actions may be taken on that screen while the modal pop-up dialog is active.

Many components now include modal pop-up dialog elements in their configuration files. Modal pop-up dialogs is enabled by default; you disable it in `app.xml`. You can also specify multiple modal pop-up window sizes in `app.xml`. Compare your custom

components to the corresponding 6.5 components to determine whether you want to include the modal pop-up dialog elements in your custom component.

UCF performance improvements in Webtop

UCF content transfer is more usable and performs better. The following lists the UCF enhancements for Documentum 6.5:

- Reduction in the number of round trips between the UCF client and server. This feature is especially effective for improving transfer performance for smaller files over a high latency WAN.
- The following UCF client initialization/startup improvements:
 - Sharing a JVM instance across multiple web sessions
 - Starting JVM upon login
- Support for PDF byte streaming through a native viewer.
- Use of parallel streams to increase content transfer rate. This feature is especially effective for improving content transfer performance of large files over a high latency WAN (outbound and inbound).

You configure this feature in `ucf.installer.config.xml`.

- Freeing up stuck threads to optimize resources and increase concurrency.
- Reduction in unnecessary WDK UCF client calls.

`residentucfinvoker` is a new UCF-related component.

Content transfer dialog

An improved content transfer dialog shows the action that is running (in the header of the dialog), the file which is transferring at the time, and progress of that transfer. The new dialog is easier to understand and is similar to other applications with which a user may be familiar.

Migrating DMCL APIs to DFC

This chapter provides information that can help you migrate a DMCL-based application to a DFC application. Please also consult the *Documentum Foundation Classes Version 6.5 Release Notes* for any known limitations or exceptions to the material in this appendix.

Overview

There are essentially three languages used to access the platform: Java, DocBasic, and C++.

If you are using Java for your customizations, they will continue to work in version 6.5. There have been no changes to the methods or interfaces of existing classes.

In previous releases, DocBasic applications accessed the DMCL via dmcl40.dll (on Windows). In version 6.5, DocBasic applications will automatically access the new dmcl.dll, which passes instructions back and forth to DFC via an emulator.

C++ accesses DMCL through dynamic links. Applications can be configured to work with dmcl40.dll, which ships with version 6.5 for backward compatibility. The applications will continue to work, but they will be working with, in essence, the 6.0 version of DMCL (with some bug fixes). C++ applications using the dmcl40.dll will not have access to methods or interfaces introduced in version 6.5 and future releases.

Methods with no corresponding DFC method

The following methods are not implemented in DFC 6.5:

- Listmessage
- Lpq
- Reset

- Unprint

Methods with corresponding DFC methods

Table 15, page 72 lists the DMCL API methods and the corresponding DFC methods. The listing is intended to help you migrate a DMCL-based application to DFC. It is not intended as a complete listing of all DFC methods.

Table 15. DMCL API methods and corresponding DFC methods

DMCL API method	DFC correspondence	
	Interface	Method name
Abort, for transactions	IDfSession	abortTrans
	IDfSessionManager	abortTransaction
Abort, for work flow	IDfWorkflow	abort
Acquire	IDfWorkItem	acquire
Addsignature	IDfSysObject	addDigitalSignature
Addsignature	IDfSysObject	addESignature
Addactivity	IDfProcess	addActivity
Addlink	IDfProcess	addLink
Addnote	IDfSysObject	addNote
	IDfPackage	appendNote
Addpackage	IDfWorkflow	addPackage
	IDfWorkitem	addPackageEx
Addpackageinfo	IDfActivity	addPackageInfo, addPackageInfoEx
Addport	IDfActivity	addPort
Addrendition	IDfSysObject	addRendition, addRenditionEx, addRenditionEx2, addRenditionEx3,
AddrouteCase	IDfActivity	addRouteCase, addConditionRouteCase
Anyevents	IDfSession	hasEvents

DMCL API method	DFC correspondence	
	Interface	Method name
Append	IDfTypedObject	appendBoolean, appendInt, appendDouble, appendId, appendString, appendTime, appendValue
Appendcontent	IDfSysObject	appendContent, appendContentEx
Appendfile	IDfSysObject	appendFile
Appendpart	IDfSysObject	appendPart
Appendstate	IDfPolicy	appendState
Apply	IDfSession, IDfQuery	apply, in IDfSession execute, in IDfQuery
Archive	IDfSession	archive
Assemble	IDfSysObject	assemble
Assume	IDfSession	assume
Attach	IDfSysObject	attachPolicy, detachPolicy
Audit	IDfAuditTrailManager	registerEventForType, registerEventForObject, registerEvents, registerEventsFromQuery, registerEventsInFolder
Authenticate	IDfClient IDfSession IDfSessionManager	authenticate
Begintran	IDfSession IDfSessionManager	beginTrans beginTransaction
Bindfile	IDfSysObject	bindFile
Branch	IDfSysObject	branch
Cachequery	IDfQuery	execute
Changepassword	IDfSession	changePassword
Checkin	IDfSysObject	checkin
Checkinapp	IDfSysObject	checkinEx

DMCL API method	DFC correspondence	
	Interface	Method name
Checkout	IDfSysObject	checkout, checkoutEx
Close	IDfCollection	close
Commit	IDfSession	commitTrans
	IDfSessionManager	commitTransaction
Complete	IDfWorkitem	complete, completeEx, completeEx2
Connect	IDfSessionManager	newSession
	IDfClient	
Count	IDfTypedObject	getAttrCount
Create	IDfSession	newObject, newObjectWithType
Createaudit	IDfAuditTrailManager	createAudit
Datatype	IDfTypedObject	getAttrDataType
Delegate	IDfWorkitem	delegateTask
Demote	IDfSysObject	demote, scheduleDemote, cancelScheduleDemote
Dequeue	IDfSession	dequeue
Dereference	IDfReplica	dereferenceReplica
	IDfMirror	dereferenceMirror
Describe	IDfSession	describe
Destroy	IDfPersistentObject	destroy
Disassemble	IDfSysObject	disassemble
Disconnect	IDfSession	disconnect (in IDfSession)
	IDfSessionManager	release (in IDfSessionManager)
Dump	IDfTypedObject	dump

DMCL API method	DFC correspondence	
	Interface	Method name
Dumpconnection	IDfSessionManager	Use getStatistics method in IDfSessionManager to return an IDfStatisticsManger object, which has the getDocbases and getSessions methods, which return information equivalent to that returned by Dumpconnection
Dumploginticket		
Encryptpass	IDfClient	encryptPassword
Execquery	IDfQuery	execute
Execsql		
Execute	IDfWorkflow	execute
Fetch	IDfSession	getObject, getObjectWith-Caching
Flush	IDfSession	flush
Flushcache	IDfSession	flushCache
Flushconnectpool	IDfSessionManager	clearIdentities
Freeze	IDfSysObject	freeze
Get	IDfTypedObject	getBoolean, getInt, getDouble, getId, getString, getTime, getValue getRepeatingBoolean, getRepeatingInt, getRepeatingDouble, getRepeatingId, getRepeatingString, getRepeatingTime, getRepeatingValue
Getconnection	IDfSessionManager	newSession
Getcontent	IDfSysObject	getContent

DMCL API method	DFC correspondence	
	Interface	Method name
Getdocbasemap	IDfDocbrokerClient	getDocbaseMap getDocbaseMapFromSpecificDocbroker
Getdocbrokermap	IDfDocbrokerClient	getDocbrokerMap
Getevents	IDfSession	getEvents
Getfile	IDfSysObject	getFile, getFileEx, getFileEx2
Getlastcoll	IDfSession	getLastCollection
Getlogin	IDfSession	getLoginTicket, getLoginTicketEx, getLoginTicketForUser
Getmessage	IDfSession	getMessage
Getpath	IDfSysObject	getPath, getPathEx, getPathEx2
Getservermap	IDfDocbrokerClient	getServerMap getServerMapFromSpecificDocbroker
Grant	IDfSysObject	grant, see also grantPermit
Halt	IDfWorkflow	halt, haltEx, haltAll
Id	IDfSession IDfTypedObject	getIdByQualification (in IDfSession) getObjectId (in IDfTypedObject)
Initcrypto	IDfClient	initCrypto
Insert	IDfTypedObject	insertBoolean, insertInt, insertDouble, insertId, insertString, insertTime, insertValue
Insertcontent	IDfSysObject	insertContent, insertContentEx
Insertfile	IDfSysObject	insertFile, insertFileEx

DMCL API method	DFC correspondence	
	Interface	Method name
Insertpart	IDfSysObject	insertPart
Insertstate	IDfPolicy	insertState
Install	IDfActivity, IDfPolicy, IDfProcess	install
Invalidate	IDfActivity, IDfPolicy, IDfProcess	invalidate
Iscached		
Kill	IDfSession	killSession (for sessions) flushObject (for SysObjects)
Link	IDfSysObject	link
Listconnection	IDfSessionManager	Use getStatistics method in IDfSessionManager to return an IDfStatisticsManager object, which has the getDochases and getSessions methods, which return information equivalent to that returned by Listconnection
Locate	IDfTypedObject	findBoolean, findInt, findDouble, findId, findString, findTime, findValue
Lock	IDfPersistentObject	lock
Mark	IDfSysObject	mark
Mount	IDfSysObject	mount
Movestate	IDfPolicy	moveState
Next	IDfCollection	next
Offset	IDfTypedObject	findAttrIndex
Pause	IDfWorkitem	pause
Print	IDfSysObject	print
Promote	IDfSysObject	promote, schedulePromote, cancelSchedulePromote
Prune	IDfSysObject	prune

DMCL API method	DFC correspondence	
	Interface	Method name
Publish_dd	IDfSession	publishDataDictionary
Purgelocal	IDfSession	purgeLocalFiles
Query_cmd	IDfQuery	execute
Query	IDfQuery	execute
Queue	IDfSysObject IDfWorkflow IDfWorkitem	queue
Readquery	IDfQuery	execute
Refresh	IDfReplica IDfMirror	refreshReplica refreshMirror
Register	IDfSysObject	registerEvent
Reinit	IDfSession	reinit
Remove	IDfTypedObject	remove
Removeactivity	IDfProcess	removeActivity
Removecontent	IDfSysObject	removeContent
Removelink	IDfProcess	removeLink
Removenote	IDfSysObject	removeNote
Removepackage	IDfWorkitem	removePackage
Removepackageinfo	IDfActivity	removePackageInfo
Removepart	IDfSysObject	removePart
Removeport	IDfActivity	removePort
Removerendition	IDfSysObject	removeRendition, removeRenditionEx, removeRenditionEx2
Removeroutecase	IDfActivity	removeRouteCase
Removestate	IDfActivity	removeState
Repeat	IDfWorkitem	repeat
Repeating	IDfTypedObject	isAttrRepeating

DMCL API method	DFC correspondence	
	Interface	Method name
Resolvealias	IDfSysObject IDfSession	resolveAlias
Restart	IDfSession IDfWorkflow	restart restartAll (for work flow)
Restore	IDfSession	restore
Resume	for lifecycles: IDfSysObject IDfworkflow IDfWorkitem	resume, scheduleResume, cancelScheduleResume (IDfSysObject) resume, resumeAll (IDfWorkflow) resume (IDfWorkitem)
Retrieve	IDfSession IDfTypedObject	getByIdByQualification (in IDfSession) getObjectId (in IDfTypedObject)
Revert	IDfPersistentObject	revert
Revoke	IDfSysObject	revoke see also revokePermit
Save	IDfPersistentObject	save
Saveasnew	IDfSysObject	saveAsNew
Seek	IDfContentCollection	seek, seekEx
Set	IDfTypedObject	setBoolean, setInt, setDouble, setId, setString, setTime, setValue setRepeatingBoolean, setRepeatingInt, setRepeatingDouble, setRepeatingId, setRepeatingString, setRepeatingTime, setRepeatingValue

DMCL API method	DFC correspondence	
	Interface	Method name
Setbatchhint	IDfSession	setBatchHint
Setcontent	IDfSysObject	setContent, setContentEx, setContentEx2
Setcontentattrs		setContentAttrs
Setdoc	IDfSysObject	setIsVirtualDocument
Setfile	IDfSysObject	setFile, setFileEx
Setoutput	IDfWorkitem	setOutput, setOutputByActivities
Setpath	IDfSysObject	setPath
Setperformers	IDfWorkflow	setPerformers
Setpriority	IDfWorkitem	setPriority
Setsupervisor	IDfWorkflow	updateSupervisorName
Shutdown	IDfSession	shutdown
Signoff	IDfPersistentObject	signoff
Suspend	IDfSysObject	suspend, scheduleSuspend, cancelScheduleSuspend
Trace	IDfSession	TraceDMCL
Truncate	IDfTypedObject	removeAll, truncate
Type	IDfSession	getTypeDescription
Unaudit	IDfAuditTrailManager	unRegisterEvent, unRegisterEventForType, unregisterEvents, unregisterEventsFromQuery, unRegisterEventsInFolder, unRegisterAllEvents
Unfreeze	IDfSysObject	unfreeze
Uninstall	IDfActivity, IDfPolicy, IDfProcess	uninstall
Unlink	IDfSysObject	unLink
Unlock	IDfSysObject	cancelCheckOut
Unmark	IDfSysObject	unMark
Unregister	IDfSysObject	unRegisterEvent

DMCL API method	DFC correspondence	
	Interface	Method name
Updatepart	IDfSysObject	updatePart, updatePartEx
Useacl	IDfSysObject	useACL
Validate	IDfActivity, IDfPolicy, IDfProcess	validate, validateProcessAndActivities
Values	IDfTypedObject	getValueCount
Vdmpath	IDfObjectPath	getAccessPath, getAccessibleFolderIds
Vdmpathdql	IDfObjectPath	getAccessPath, getAccessibleFolderIds
Verifyaudit	IDfPersistentObject	verifyAudit
Verifiesignature	IDfSysObject	verifySignature

Object Type and Property Changes for version 6.5

These tables describe types and properties that are new, changed, deprecated, or obsolete in version 6.5.

New object types

[Table 16, page 83](#) lists the new object types for version 6.5.

Table 16. New Object Types

Type Name	Description
dm_bocs_config	Records configuration information for a BOCS server
dm_client_registration	Records a client instance's registration, for use in authorizing permission and privilege escalation requests
dm_client_rights	Records the privileged roles a client instance is allowed to assert.
dm_cont_transfer_config	Defines content transfer capabilities for a distributed environment
dmc_class	Added with Smart Container DAR file.
dmc_constraint_set	Added with Smart Container DAR file.
dmc_metamodel	Added with Smart Container DAR file.
dm_dms_config	Records configuration information for a DMS server

Type Name	Description
dm_lightweight	Serves as a supertype for all user-defined lightweight object types. This is a 'pseudo' type in that there are no repository tables for this type and instances of this type cannot be created.
	This type is only for internal use in 6.5.
dm_message_route_user_data	Records route-specific user information for an email message
dm_validation_descriptor	Used internally in distributed environments
dmc_preset_package	Installed with the Preset DAR file during repository configuration. Used by WDK-based applications
dmc_preset_info	Installed with the Preset DAR file during repository configuration. Used by WDK-based applications
dmc_relationship_def	Defines a relationship. This provides a higher level of abstraction for relationships, allowing easier management and querying of relationships by applications.
dmc_scope_config_relation	Installed with the Preset DAR file during repository configuration. Used by WDK-based applications
The following object types are added to support a new workflow feature:	These types support the ability to add structured data to a workflow.
dmc_wfsd_element	
dmc_wfsd_element_boolean	
dmc_wfsd_element_date	
dmc_wfsd_element_double	
dmc_wfsd_element_integer	
dmc_wfsd_element_string	
dmc_wfsd_parent	
dmc_type_info	
dmc_wfsdrp_boolean	
dmc_wfsdrp_date	
dmc_wfsdrp_double	
dmc_wfsdrp_integer	

Type Name	Description
dmc_wfsdrp_string dmc_wfsdrp_parent	These types are installed with the Collaboration Services DocApp.
The following types are added to support Collaboration Services:	
dmc_calendar dmc_calendar_event dmc_datatable dmc_datatable_row dmc_datatable_ schema dmc_datatable_ settings	
dmc_xfm_adaptor_ config	Records information about a forms adaptor.

Changed object types

[Table 17, page 85](#) lists the changes to existing types.

Table 17. Changed Object Types

Type Name	Description
dm_acs_config	<p>Added the following properties:</p> <ul style="list-style-type: none"> server_major_version server_minor_version <p>Modified the following properties:</p> <ul style="list-style-type: none"> acs_rw_capability is_cache_acs

Type Name	Description
dm_activity	<p>Added the following properties:</p> <ul style="list-style-type: none"> • activity_group_flag • activity_group_id • exec_retry_interval • exec_retry_max • post_timer_calendar_flag • post_timer_calendar_id • pre_timer_calendar_flag • pre_timer_calendar_id • r_performer_cond_id • r_performer_cond_name • r_performer_cond_user • r_port_type, from string(8) to string(16) • sd_element_flag • sd_element_name <p>The following string properties were lengthened to 128:</p> <ul style="list-style-type: none"> • r_package_name • resolve_pkg_name <p>The following properties have new values added to their valid values:</p> <ul style="list-style-type: none"> • exec_err_handling • exec_subtype
dm_aspect_type	<p>Added the following properties:</p> <ul style="list-style-type: none"> • i_attr_def • is_data_shared
dm_audittrail	<p>Added the following property:</p> <ul style="list-style-type: none"> • attribute_list_old
dmi_audittrail_attrs	<p>Added the following property:</p> <ul style="list-style-type: none"> • attribute_list_old
dmc_completeded_workitem	<p>Lengthened the following property:</p> <ul style="list-style-type: none"> • act_name

Type Name	Description
dmr_content	Added the following properties: <ul style="list-style-type: none"> • i_parked_state • other_file_size
dm_dd_info	Added the following property: <ul style="list-style-type: none"> • fulltext_support <p>Note: Used only for lightweight object subtypes.</p> <p>The reference_kind property, previously unused, is now used by Collaboration Services for internal purposes.</p>
dm_dd_attr_info	Added the following new property: <ul style="list-style-type: none"> • ftindex_attrs <p>Note: Used only for lightweight object subtypes.</p> <p>The reference_kind property, previously unused, is now used by Collaboration Services for internal purposes.</p>
dm_docbase_config	Added the following properties: <ul style="list-style-type: none"> • audit_old_values • docbase_roles • r_normal_tz • approved_clients_only
dm_format	Added the following property: <ul style="list-style-type: none"> • a_page_plugin_name <p>Modified the following property:</p> <ul style="list-style-type: none"> • mime_type This property was lengthened from 64 to 256 in 5.3 SP5.
dm_func_expr	Lengthened the following property: <ul style="list-style-type: none"> • object_alias
dm_group	Added the following properties: <ul style="list-style-type: none"> • i_nondyn_supergroups_names • is_module_only • is_protected

Type Name	Description
dm_ldap_config	Added the following properties: <ul style="list-style-type: none">• failover_ldap_config_ids• failover_use_interval• map_rejection• retry_count• retry_interval Modified the following properties: <ul style="list-style-type: none">• map_val_typeAdded support for the value "E" (meaning expression) for this property
dm_media_profile	Added the following properties: <ul style="list-style-type: none">• filter_names• filter_values• related_objects_only• src_obj_type
dm_message_address	The addr_type property was lengthened from string(1) to string(2)
dm_message_route	The following property was added: <ul style="list-style-type: none">• route_user_data_hash_id
dmc_module	Added the following properties: <ul style="list-style-type: none">• a_is_privileged• a_privilege_roles
dm_process	Added the following properties: <ul style="list-style-type: none">• act_performer_from• act_performer_rule• act_performer_to• calendar_id• execution_flag• post_timer_calendar_flag• post_timer_calendar_id• pre_timer_calendar_flag• pre_timer_calendar_id• sd_element_acl• sd_element_default_acl• sd_element_default_value

Type Name	Description
	<ul style="list-style-type: none"> sd_element_name sd_element_options sd_element_parent_id sd_element_type <p>The following string properties were lengthened to 128:</p> <ul style="list-style-type: none"> act_choose_by act_choose_for act_choose_name r_link_dest_act r_link_src_act
dm_public_key_certificate	<p>Modified the following properties:</p> <ul style="list-style-type: none"> key_type private_key_identifier
dmi_queue_item	<p>Lengthened the following property:</p> <ul style="list-style-type: none"> task_name
dm_relation_type	<p>Added the following property:</p> <ul style="list-style-type: none"> a_controlling_kind
dm_retainer	<p>Added the following property:</p> <ul style="list-style-type: none"> aging_method
dmc_routecase_condition	<p>Lengthened the following property:</p> <ul style="list-style-type: none"> a_object_alias
dm_smart_list	<p>Added the following new properties:</p> <ul style="list-style-type: none"> has_results query_type results_count selected_sources
dm_sysobject	<p>The a_extended_properties property is no longer used to record the object ID of the room governing the object if the object is in a room. A SysObject's governing room is now recorded in the a_gov_room_id property, which is an aspect property associated with the SysObject when the Sysobject is placed in the room.</p>
dmc_transition_condition	<p>Lengthened the following property:</p> <ul style="list-style-type: none"> r_object_alias

Type Name	Description
dm_type	Added the following new properties: <ul style="list-style-type: none">• attr_identifier• attr_restriction• next_attr_identifier• type_category
dmi_type_info	Added the following new properties: <ul style="list-style-type: none">• i_type_features• type_version• default_aspects
dmc_aspect_type	Added the following new properties: <ul style="list-style-type: none">• i_attr_def• is_data_shared
dm_workflow	Added the following new properties: <ul style="list-style-type: none">• correlation_identifer• initiate_act• parent_act_name• parent_act_seqno• parent_id <p>The following string properties were lengthened to 128:</p> <ul style="list-style-type: none">• r_act_name• r_perf_act_name
dmi_workitem	Added the following properties: <ul style="list-style-type: none">• r_exec_retried_count• r_handling_instruction• r_next_retry_date• r_target_task_id <p>The following string property was lengthened to 128:</p> <ul style="list-style-type: none">• r_runtime_state
dmc_workqueue	Lengthened the following string property to 128: <ul style="list-style-type: none">• package_name
dmc_wf_package_skill	Added the following new property: <ul style="list-style-type: none">• skill_info_ids

Type Name	Description
dmi_wf_timer	<p>Added the following new properties:</p> <ul style="list-style-type: none"> • r_calendar_id • r_calendar_status <p>Lengthened the following string property to 128:</p> <ul style="list-style-type: none"> • r_act_name
dmc_workqueue_policy	<p>Added the following new properties:</p> <ul style="list-style-type: none"> • calendar_id • increment_priority_method • increment_priority_mode

Deprecated or obsolete properties

[Table 18, page 91](#) lists the persistent properties that are either deprecated or obsolete as of version 6.5.

Table 18. Deprecated and obsolete properties

Object type	Property	Deprecated or obsolete
session config	docbase_scope	obsolete

[Table 19, page 91](#) lists the computed properties that are either deprecated or obsolete as of version 6.5.

Table 19. Deprecated and obsolete computed properties

Object type	Property	Deprecated or obsolete
_content_buffer	S	obsolete

Properties added conditionally

The properties listed in [Table 20, page 92](#) are only added to an object type under certain conditions. The description of each property explains the property's use and the conditions under which it appears in a object type definition.

Table 20. Properties added conditionally

Property	Datatype	Single/ repeating	Description
i_shared_status	integer	S	<p>This property is added to the object type definitions of types that are created as shareable types. Only dm_sysobject or its subtypes can be shareable types.</p> <p>Note: Shareable types are currently only used internally. Users and applications cannot create shareable types.</p> <p>The property indicates whether an instance of the object type is shared by any lightweight object. Valid values are:</p> <p>0, the object is not shared</p> <p>1, the object is shared</p>
i_property_bag	string(2000)	S	<p>This property is added to an object type if the type definition contains a NONQUALIFIABLE property. The property can also be explicitly added by altering the object type.</p> <p>The property stores the names and values of NONQUALIFIABLE properties. It is also used to store the names and values of aspect properties if the properties are added</p>

Property	Datatype	Single/ repeating	Description
r_property_bag	string(2000)	S	<p>to the aspect with the OPTIMIZEFETCH option.</p> <p>This property is added to an object type if the type definition contains a NONQUALIFIABLE property. Altering a type to add i_property_bag also adds r_property_bag automatically.</p> <p>This property stores any overflow from i_property_bag.</p>

Deprecated or obsolete object types

[Table 21, page 93](#) lists the changes to existing types.

Table 21. Deprecated or Obsolete Object Types

Type Name	Deprecated	Obsolete
dmi_linkrecord	X	
dm_linked_store	X	
dm_router		X
enable_workitem_mgmt	Boolean	Previously used to enable workqueue use

Changed properties

The implementation of the local_diskfull_limit property in the non-persistent objects that define a session configuration's configuration has changed. Previously, the limit specified in this property was expressed as a percentage. For version 6.5, the limit value is now expressed as a number of megabytes.

Valid values are now from 0 and 100. Values from 1 to 100 are interpreted as megabytes. For example, a value of 50 means that the client local area is limited to 50MB in size. A value of 0 means that the size is unlimited.

Configuration Settings in WDK-based Application Deployment

These tables list the mandatory and optional configuration elements that can be set before, during, and after deployment of WDK-based applications. Because WDK-based applications encapsulate DFC, you can also configure DFC settings as described in `dfcfull.properties`. This file is located in the `WEB-INF/classes` directory of the WDK-based application. Functions marked with an asterisk (*) must be performed for every deployment.

[Table 22, page 95](#) lists the configuration elements that must be set before deploying a WDK-based application, such as Webtop or TaskSpace. Not all of these elements must be set for every deployment, but if you wish to support the function in the first column, you must enable it before deployment. Refer to *Web Development Kit and Webtop Deployment Guide* for more information on these settings.

Table 22. Mandatory configuration before deployment

Function	Element	Location
Turn off tag pooling (Tomcat, Oracle)*	<code>servlet.init-param</code>	<code>web.xml</code>
Global registry indications*	<code>dfc.docbroker.host</code> <code>dfc.globalregistry.repository</code> <code>dfc.globalregistry.username</code> <code>dfc.globalregistry.password</code>	<code>dfc.properties</code> in <code>WEB-INF/classes</code>
WAS compiler and classloader*	<code>Classloader order</code> <code>useJDKCompiler</code>	WAS admin console

[Table 23, page 95](#) lists the optional configuration settings that can be set before deployment.

Table 23. Optional configuration before deployment

Function	Element	Location
WAS failover	<code>NoAffinitySwitchBack</code>	WAS cluster configuration

Function	Element	Location
WAS global security	Security policies and environment variables	Download from Powerlink if needed
UCF to use file, not Windows registry	registry.mode	ucf.installer.config.xml
Default content transfer directories for client	option	ucf.installer.config.xml
Unsigned SSL certificates	option	ucf.installer.config.xml
Proxy servers	http11.chunked.transfer	ucf.server.config.xml in WEB-INF/classes
Set content transfer mode	contentxfer.default-mechanism	custom/app.xml
Change ACS and BOCS behavior	contentxfer.accelerated-read contentxfer.accelerated-write	custom/app.xml
Locale settings	language	custom/app.xml
Java EE principal authentication	principal credentials securityconstraint	TrustedAuthenticator-Credentials.properties in WEB-INF/classes/com/documentum/web/formext/session web.xml in WEB-INF

Table 24, page 96 lists the configuration settings that can be changed after deployment.

Table 24. Optional configuration after deployment

Function	Element	Location
Turn off failover	failover.enabled	custom/app.xml
Change presets repository (default = global registry)	presets	custom/app.xml
Change preferences repository (default = global registry)	preferencesrepository	custom/app.xml
Default repository	authentication.docbase	custom/app.xml
Allow saved credentials	save_credentials	custom/app.xml
Encrypt passwords for drl, drlauthenticate, and virtuallinkconnect		com.documentum.web.formext.session.TrustedAuthenticatorTool

Function	Element	Location
Enable accessibility	accessibility	custom/app.xml
Configure app server timeout	sessionconfig	web.xml in WEB-INF

dfc.properties

These topics are included:

- [Overview, page 99](#)
- [Changes to existing key names, page 99](#)
- [dmcl.ini key migration to dfc.properties, page 102](#)
- [Obsolete dmcl.ini and session configuration options, page 104](#)
- [Obsolete dfc.properties keys, page 106](#)

Overview

In version 6, DFC replaced the Server API as the API for Content Server. As part of this change, the dmcl.ini file became obsolete and its relevant entries were migrated to the dfc.properties file. In addition, the naming conventions for entries in the dfc.properties file were standardized. This appendix describes the changes to the dfc.properties file.

Changes to existing key names

[Table 25, page 99](#), describes the changes to existing key names. Both new and old names are listed. For backward compatibility, both new and old names continue to work in version 6.5. Invalid entries do not generate an error, but have no effect on functionality.

Table 25. Name changes for existing dfc.properties for version 6.5

Old name	New name
dfc.acs.avail.refresh.frequency	dfc.acs.avail.refresh_interval
dfc.acs.config.refresh.frequency	dfc.acs.config.refresh_interval

Old name	New name
dfc.acs.network_location.refresh.frequency	dfc.acs.network_location.refresh_interval
dfc.admin.ldif.file.charset	dfc.admin.ldif_file_charset
dfc.appledouble.resource.file.extension	dfc.appledouble.resource_file_extension
dfc.cacs.check.http.method	dfc.bocs.check.http_method
dfc.cacs.check.keep.number	dfc.bocs.check.keep_number
dfc.bocs.config.refresh.frequency	dfc.bocs.config.refresh_interval
dfc.cache.append.name	dfc.bof.cache.append_name
dfc.bof.cacheconsistency.interval	dfc.bof.cache.currency_check_interval
dfc.bof.registry.connect.attempt.interval	dfc.globalregistry.connect_attempt_interval
dfc.bof.registry.preload.enabled	dfc.bof.cache.enable_preload
dfc.bof.registry.password	dfc.globalregistry.password
dfc.bof.registry.repository	dfc.globalregistry.repository
dfc.bof.registry.username	dfc.globalregistry.username
dfc.cache.ddinfo.globalCacheSize	dfc.cache.ddinfo.size
dfc.cache.dir	dfc.cache_dir
dfc.cache.object.globalCacheSize	dfc.cache.object.size
dfc.cache.query.globalCacheSize	dfc.cache.query.size
dfc.core.truncate_long_values	dfc.compatibility.truncate_long_values
dfc.config.timeout	dfc.config.check_interval
dfc.checkout.dir	dfc.data.checkout_dir
dfc.data.dir	dfc.data.dir
dfc.docbase.max_deadlock_retries	dfc.session.max_deadlock_retries
dfc.docbase.max_error_retries	dfc.session.max_error_retries
dfc.exception.include_decoration	No change
dfc.exception.include_id	No change
dfc.exception.include_stack	No change
dfc.export.dir	dfc.data.export_dir
dfc.housekeeping.cleanup.interval	dfc.resources.cleanup_interval
dfc.max.vdm.children.flush.count	dfc.vdm.max_children_flush_count

Old name	New name
dfc.recordInlineDescendants	dfc.xml.record_inline_descendants
dfc.registry.file	No change
dfc.registry.mode	No change
dfc.resources.diagnostics.enabled	dfc.diagnostics.resources.enable
dfc.search.docbase.brokers	dfc.search.docbase.broker_count
dfc.search.ecis.adapter.domain	No change
dfc.search.ecis.brokers	dfc.search.ecis.broker_count
dfc.search.ecis.enable	No change
dfc.search.ecis.host	No change
dfc.search.ecis.password	No change
dfc.search.ecis.port	No change
dfc.search.ecis.access.timeout	dfc.search.ecis.request_timeout
dfc.search.ecis.name	dfc.search.ecis.rmi_name
dfc.search.ecis.login	dfc.search.ecis.username
dfc.search.formatcache.timeout	dfc.search.formatcache.refresh_interval
dfc.search.fulltext.enabled	dfc.search.fulltext.enable
dfc.search.ecis.max.results	dfc.search.ecis.max_results
dfc.search.sourcecache.timeout	dfc.search.sourcecache.refresh_interval
dfc.search.typecache.timeout	dfc.search.typecache.refresh_interval
dfc.session.dynamic_delay	No change
dfc.session.pool.enabled	dfc.session.pool.enable
dfc.session.pool.timeout	dfc.session.pool.expiration_interval
dfc.session.surrogate.check.interval	dfc.session.surrogate.check_interval
dfc.session.surrogate.mode	No change
dfc.storagepolicy.diagnostics.enabled	dfc.diagnostics.storagepolicy.enable
dfc.storagepolicy.ignore.rule.errors	dfc.storagepolicy.ignore_rule_errors
dfc.storagepolicy.validation.interval	dfc.storagepolicy.validation_interval
dfc.strictURI	dfc.xml.use_strict_uri
dfc.tracing.baseTraceFileName	dfc.tracing.file_prefix
dfc.tracing.compactModeBufferSize	dfc.tracing.compact_mode_buffer_size
dfc.tracing.enabled	dfc.tracing.enable

Old name	New name
dfc.tracing.entrancePointExprs	dfc.tracing.method_name_filter
dfc.tracing.loggingMode	dfc.tracing.file_creation_mode
dfc.tracing.maxFileSize	dfc.tracing.max_file_size
dfc.tracing.maxThreadsToTrace	dfc.tracing.max_threads_to_trace
dfc.tracing.maxUsersToTrace	dfc.tracing.max_users_to_trace
dfc.tracing.mode	No change
dfc.tracng.rpcCountingEnabled	dfc.tracing.display_rpc_count
dfc.tracing.scriptableMethodsMarked	dfc.tracing.display_scriptable_mark
dfc.tracing.stackDepth	dfc.tracing.max_stack_depth
dfc.tracing.threadNameExprs	dfc.tracing.thread_name_filter
dfc.tracing.timestampDateFormat	dfc.tracing.date_format
dfc.tracing.timingStyle	dfc.tracing.timing_style
dfc.tracing.traceFileDirectory	dfc.tracing.dir
dfc.tracing.userNameExprs	dfc.tracing.user_name_filter
dfc.user.dir	dfc.data.user_dir
dfc.validation.expr.currency.check	dfc.validation.expr.currency_check_interval
dfc.validation.expr.debug.all	No change
dfc.validation.expr.debug.code	No change
dfc.validation.expr.debug.eval	No change
dfc.validation.expr.debug.tree	No change
dfc.validation.expr.disable_java	No change
dfc.validation.overrides.currency.check	dfc.validaton.overrides.currency_check_interval

dmcl.ini key migration to dfc.properties

[Table 26, page 103](#) describes the dmcl.ini keys that migrated to the dfc.properties file in version 6.5.

Table 26. dfc.properties keys migrated from dmcl.ini file

dmcl.ini key	Corresponding new dfc.properties key
application_code	dfc.application_code
auto_request_forward	dfc.docbroker.auto_request_forward
batch_hint_size	dfc.batch_hint_size
backup_host	dfc.docbroker.host
backup_port	dfc.docbroker.port
backup_protocol	dfc.docbroker.protocol
backup_service	dfc.docbroker.service
backup_timeout	dfc.docbroker.timeout
castore_write_max_attempts	dfc.content.castore.write_max_attempts
castore_write_sleep_interval	dfc.content.castore.write_sleep_interval
client_date_format	dfc.date_format
client_locale	dfc.locale
connect_pooling_enabled	dfc.session.pool.enable
connect_retry_interval	dfc.session.connect_retry_interval
connect_retry_limit	dfc.session.connect_retry_limit
debug_dbid	dfc.docbroker.debug.docbase_id
debug_host	dfc.docbroker.debug.host
debug_port	dfc.docbroker.debug.port
debug_service	dfc.docbroker.debug.service
docbroker_search_order	dfc.docbroker.search_order
ini_file_path	dfc.config.file
local_clean_on_init	dfc.data.local_clean_on_init
local_diskfull_check	dfc.data.diskfull_check
local_diskfull_limit	dfc.data.diskfull_limit
local_path	dfc.data.local_dir
local_purge_on_diskfull	dfc.data.local_purge_on_diskfull
max_collection_count	dfc.session.max_collection_count
max_session_count	dfc.session.max_session_count
primary_host	dfc.docbroker.host
primary_port	dfc.docbroker.port

dmcl.ini key	Corresponding new dfc.properties key
primary_protocol	dfc.docbroker.protocol
primary_service	dfc.docbroker.service
primary_timeout	dfc.docbroker.timeout
re_binding_label	dfc.reference.binding_label
secure_connect_default	dfc.session.secure_connect_default
token_storage_path	dfc.tokenstorage.dir
token_storage_enabled	dfc.tokenstorage.enable
umask	dfc.data.umask
use_compression	dfc.content.use_compression
use_content_server	dfc.content.use_content_server

Obsolete dmcl.ini and session configuration options

Table 27, page 104 lists the dmcl.ini keys that are obsolete in version 6.5 and have no equivalent to set in dfc.properties. It also lists properties formerly present in the session configuration objects that are obsolete in version 6.5.

Table 27. Obsolete session configuration options

Entry	Source	Comments
block_during_rpc	dmcl.ini	Is specific to native code DMCL.
client_codepage	dmcl.ini	none
client_os_codepage	dmcl.ini	none
connect_callback_enabled	api config and session config objects	none
connect_failure_callback	api config and session config objects	none
connect_failure_data	api config and session config objects	none
connect_success_callback	api config and session config objects	none

Entry	Source	Comments
connect_success_data	api config and session config objects	none
content_callback_data	api config and session config objects	none
content_callback_function	api config and session config objects	none
local_diskfull_warn	dmcl.ini	none
network_callback_data	api config and session config objects	none
network_callback_function	api config and session config objects	none
new_connection_callback		none
new_connection_data		none
nfs_enabled	dmcl.ini	
r_trace_file	dmcl.ini	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.
r_trace_level	dmcl.ini	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.
client_cache_size	dmcl.ini	Implementation now allows per-session caches to dynamically adapt to free memory.
connect_timeout	dmcl.ini	Is specific to native code DMCL.
connect_recycle_interval	dmcl.ini	Is specific to native code DMCL.
exception_count		Is specific to native code DMCL.
exception_count_interval		Is specific to native code DMCL.

Entry	Source	Comments
terminate_on_exception		Is specific to native code DMCL.
i_override_list		
cache_queries		
max_connection_per_session		
use_local_always	dmcl.ini	Option to use server common area is not available in DFC 6.5, so this becomes unneeded
use_local_on_copy	dmcl.ini	Option to use server common area is not available in DFC 6.5, so this becomes unneeded

Obsolete dfc.properties keys

[Table 28, page 106](#) lists the dfc.properties keys that are obsolete in version 6.5. Setting these keys have no effect on DFC 6.5.

Table 28. Obsolete dfc.properties keys

Entry	Source	Comments
dfc.tracing.combineDMCL	dfc.properties	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.
dfc.tracing.compactMode	dfc.properties	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.

Entry	Source	Comments
dfc.tracing.recordParameters	dfc.properties	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.
dfc.tracing.recordReturn-Value	dfc.properties	Replaced by new tracing implementation—refer to the <i>Content Server Administrator's Guide</i> for information.

A

- attribute, 27
- auto complete, 61

B

- BOF2 modules
 - migrating, 53

C

- CHANGE...OBJECT statement, 30
- CLEAN_LINKS, 28
- column resizing, 64
- configuration
 - WDK, 95
- configuration service extensions, 61
- consistency checker utility, 50
- Content Server
 - listener queue length, configuring, 31
- content store
 - changing location, 15
- content transfer applet, 59
- ctrl-click, 64

D

- data grid enhancements, 62
- database
 - changing location, 15
 - changing version, 15
- datacolumnbeginswith, 61
- date literals, 30
- dfc.compatibility.truncate_long_
values, 27
- dfc.machine.id, 27
- dfc.properties, 27 to 29
- dfc.session.allow_trusted_login, 29
- dm_bof_registry, 37
- dm_extern_file, 28
- dm_linkedstore, 28

- dm_startedworkitem, 28
- dmcl.ini-keys, 29
- dmi_linkrecord, 28
- DocApps
 - migrating, 53
- DocList component, 64
- DQL
 - CHANGE...OBJECT statement, 30
 - date literals, 30
 - migrating content, 52
 - POSITION keyword, 30
- drag and drop, 66
- dump and load, 15
- dynamic filters, 61

E

- enable_workitem_mgmt (server.ini
key), 31

F

- fixed column headers, 65
- fixed menus, 59

G

- global registry
 - copying, 50
 - defined, 36

H

- hotkeys, 60

I

- IDfSession.setServerTraceLevel, 31
- insertbefore, 62
- installation order
 - new system, 13

K

keyboard shortcuts, 60

L

LDIF file, 29
lifecycle, 65
lightweight sysobjects
 migrating to, 26
linked store storage areas, 28
listener_queue_length, 31
listener_queue_length (server.ini key), 31
login tickets
 backwards compatibility, 51
LWSOs
 migrating to, 26

M

mainex component, 58
max_backup_index, 29
max_file_size, 29
memory
 application server usage, 18
 DFC usage, 18
migrating
 BOF2 and DocApps, 53
migrating content with DQL, 52
migration
 overview, 12
modules
 migrating, 53
multi-object select, 64

O

optical storage devices, 28

P

performance
 planning, 18
 Server, common problems, 19
 web application, 20
planning worksheet
 application server host, 34
Planning worksheet
 client machine, 35
 Content Server host, 33
 customized components, 35
 index server host, 34

POSITION keyword, 30
preference persistence, 66
presets, 57

Q

query
 performance enhancement, 20

R

relationships, 65
resizable columns, 64
right-click menus, 59
row selection, 62

S

safe harbor, 9
server.ini file
 enable_workitem_mgmt key, 31
server_login_ticket_version (server.ini
 key), 52
shift-click, 64
showifinvalid, 62
showinvalidactions, 62
Streamline, 58
SYNC_REPLICA_RECORDS, 29
system
 sizing, 17
system updates
 order, 13

T

tab order, 60
themes, 58
Toolbar component, 59
tracing, memory
 DFC, 18

U

upgrade
 overview, 11

W

work flow
 enable_workitem_mgmt (server.ini
 key), 31