EMC® Documentum® System

Version 7.2

Upgrade and Migration Guide
Legal Notice

Copyright © 1994-2016 EMC Corporation. All Rights Reserved.

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. Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries. All other trademarks used herein are the property of their respective owners.

Documentation Feedback

Your opinion matters. We want to hear from you regarding our product documentation. If you have feedback about how we can make our documentation better or easier to use, please send us your feedback directly at ECD.Documentation.Feedback@emc.com
## Table of Contents

### Preface

<table>
<thead>
<tr>
<th>Table of Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
</tr>
</tbody>
</table>

### Chapter 1 Upgrade and Migration Overview

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade and Migration</td>
<td>13</td>
</tr>
<tr>
<td>Understanding Migration</td>
<td>14</td>
</tr>
<tr>
<td>Order of New Product Installation</td>
<td>15</td>
</tr>
<tr>
<td>Order of System Updates</td>
<td>16</td>
</tr>
</tbody>
</table>

### Chapter 2 Planning the Documentum System Upgrade

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Upgrade Strategies</td>
<td>19</td>
</tr>
<tr>
<td>Changes that Impact Content Server Upgrade or Migration</td>
<td>22</td>
</tr>
<tr>
<td>Linking to the 32-bit Oracle Client Library on Linux</td>
<td>22</td>
</tr>
<tr>
<td>Upgrading Multiple Repositories to the Same Version</td>
<td>23</td>
</tr>
<tr>
<td>Mapping Your Current Configuration</td>
<td>24</td>
</tr>
<tr>
<td>Designing a Documentum 7.x Configuration</td>
<td>25</td>
</tr>
<tr>
<td>Addressing Hardware Concerns</td>
<td>26</td>
</tr>
<tr>
<td>Upgrading Third-party Software</td>
<td>27</td>
</tr>
<tr>
<td>Planning for Global Registries</td>
<td>27</td>
</tr>
<tr>
<td>Mapping a Documentum 7.x Configuration</td>
<td>28</td>
</tr>
<tr>
<td>Planning Upgrade and Migration to Documentum 7.x</td>
<td>28</td>
</tr>
<tr>
<td>Setting Up a Test Environment</td>
<td>29</td>
</tr>
<tr>
<td>Creating the Test Environment</td>
<td>29</td>
</tr>
<tr>
<td>Migrating a TCS-Enabled Docbase to a New Server</td>
<td>30</td>
</tr>
<tr>
<td>Client-First Migration</td>
<td>30</td>
</tr>
<tr>
<td>Planning Upgrade for Repositories in a Federation</td>
<td>31</td>
</tr>
<tr>
<td>Guidelines for Upgrading a Distributed Configuration</td>
<td>31</td>
</tr>
<tr>
<td>Planning the Upgrade or Migration of the AEK Key to Lockbox</td>
<td>31</td>
</tr>
<tr>
<td>Enabling Remote Key Management</td>
<td>31</td>
</tr>
</tbody>
</table>

### Chapter 3 Upgrading Content Server

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade Checklist</td>
<td>33</td>
</tr>
<tr>
<td>Changing the Database Operating System and Version</td>
<td>35</td>
</tr>
<tr>
<td>Preparing DB2 databases for Upgrade</td>
<td>36</td>
</tr>
<tr>
<td>Migrating the Database to UTF-8</td>
<td>36</td>
</tr>
<tr>
<td>The database_refresh_interval Key</td>
<td>36</td>
</tr>
<tr>
<td>Ensuring the Completion of Automatic Tasks Before Upgrading</td>
<td>36</td>
</tr>
<tr>
<td>Ensuring that the dm_server_config Object is Unlocked</td>
<td>37</td>
</tr>
<tr>
<td>Upgrading to 64-bit Content Server</td>
<td>37</td>
</tr>
<tr>
<td>64-bit ODBC and DSN Libraries</td>
<td>38</td>
</tr>
<tr>
<td>Errors During Upgrade</td>
<td>39</td>
</tr>
<tr>
<td>Components Not Supported after Upgrade</td>
<td>39</td>
</tr>
<tr>
<td>Migrating Custom Plug-ins</td>
<td>40</td>
</tr>
<tr>
<td>Upgrading Content Server on Red Hat Linux 5.x Systems</td>
<td>40</td>
</tr>
</tbody>
</table>
# Table of Contents

Upgrading the Content Server Software ................................................. 40
Upgrading Content Server Installed with Cluster Services ..................... 44
Upgrading Content Server in a Distributed or Load-Balanced Configuration .......................................................... 45
  Upgrading the BOCS Server .......................................................... 47
  Upgrading the DMS Server .......................................................... 47
Post-Upgrade Tasks ............................................................................. 48
  Improving Performance after Upgrade .............................................. 48
  Extending the Oracle Tablespace Size .............................................. 48
  Enhancements after Upgrade .......................................................... 49
  Reinstalling D2 Client Libraries ...................................................... 49
  Rebuilding the database views ......................................................... 49

## Chapter 4 Upgrade Scenarios ................................................................. 51
Upgrading Content Server 6.5 SP2 to 7.2 — Windows/SQL Server ............ 51
Upgrading Content Server 6.5 SP3 to 7.2 — AIX/DB2 ........................... 55
Upgrading Content Server 6.5 SP3 to 7.2 — Windows/SQL Server .......... 59
Upgrading Content Server 6.7 to 7.2 — Windows/Oracle ......................... 63
Upgrading Content Server 6.7 SP1 to 7.2 — Linux/Oracle ......................... 64
Upgrading Content Server 6.7 SP2 to 7.2 — Solaris/Oracle ....................... 68
Upgrading Content Server 7.0 to 7.2 — Windows/SQL ........................... 72
Upgrading Content Server 7.1 to 7.2 — Linux/Oracle ........................... 75

## Chapter 5 Migrating Content Server ....................................................... 77
Understanding the Migration Process ..................................................... 77
Migration Checklist ............................................................................. 79
Planning a Migration ........................................................................... 80
Migration Methods .............................................................................. 81
  Method 1: Migrating a Repository .................................................. 82
  Method 2: Copying a Repository ..................................................... 83
Migrating Data Using SQL Server ......................................................... 85
Consolidating Repositories .................................................................. 86
Migrating Data from Earlier Versions of Content Server ......................... 90
Migrating XML Content to the XML Store ........................................... 90
Using DQL to Migrate Content to an XML Store .................................... 91
Migrating Custom Content Server Methods ......................................... 91
Migrating DocApps and BOF2 Modules ............................................... 91
Post-Migration Tasks .......................................................................... 92

## Chapter 6 Migrating DFC Customizations ................................................. 93
Java Class Changes .............................................................................. 93
Configuring DFC for Native IPv4 Operations ......................................... 94
Configuring 6.7 Clients to Work with Content Server 7.x ....................... 94
Migrating Customizations to Business Objects ....................................... 94
Migrating DMCL API Calls to DFC API Calls ........................................ 95
Search Service .................................................................................... 95
Full Format Specifications No Longer Accepted .................................... 95
Chapter 7  Migrating DFS Customizations .......................................................... 99
   Upgrading the DFS .NET Productivity Layer ............................................. 99
      Upgrading from a Version Earlier Than 7.x ........................................ 100
      Upgrading from a Pre–7.x Minor Version or Service Pack ................. 101
   Restoring Trusted Certificates after Upgrading UCF .......................... 101
   Trusted Login is Disabled By Default .................................................. 102
   Cookie Consistency Check ............................................................... 102
   .NET Framework Update .................................................................. 102

Chapter 8  Migrating CMIS Customizations ..................................................... 103
   getFolderParentReturnsFeed .................................................................. 103

Chapter 9  Migration Scenarios ................................................................... 105
   Migrating Content Server 6.5 SP2 to 7.2 — Windows/SQL Server .......... 105

Appendix A  Migrating DMCL APIs to DFC ................................................. 111
   Overview ......................................................................................... 111
   Methods with no corresponding DFC method .................................... 111
   Methods with corresponding DFC methods ..................................... 112

Appendix B  Object Type and Property Changes ..................................... 119
   New object types ............................................................................. 119
   Changed object types ..................................................................... 120
   Changed object types with new properties ..................................... 127

Appendix C  dfc.properties ........................................................................ 129
   Overview ......................................................................................... 129
   Changes to existing key names ......................................................... 129
   dmcl.ini key migration to dfc.properties ...................................... 132
   Obsolete dmcl.ini and session configuration options ..................... 133
   Obsolete dfc.properties keys ......................................................... 135

Appendix D  LDAP Certificates ................................................................. 137
List of Figures

Figure 1. System Installation Order, New Documentum System ........................................ 15
Figure 2. System Update Order, Existing Documentum System ........................................ 16
Figure 3. System Upgrade Scenarios ................................................................................. 21
Figure 4. Upgrade Steps from 32-bit Content Server to 64-bit Content Server ................... 38
Figure 5. Pre-7.x 32-bit Content Server Environment .................................................... 78
Figure 6. Migration Process .............................................................................................. 79
# List of Tables

Table 1. Upgrade and Migration of Product Components ........................................ 13  
Table 2. Content Server and Database Server Host Worksheet ................................... 23  
Table 3. Application Server Host Worksheet .................................................................. 24  
Table 4. Index Server Host Worksheet ........................................................................... 24  
Table 5. Client Machine Worksheet .............................................................................. 25  
Table 6. Customized Components Worksheet .................................................................. 25  
Table 7. Memory Consumption by Documentum Executables ......................................... 26  
Table 8. Parameters required by dm_acs_install_ebs script ........................................... 46  
Table 9. Base and Upgraded OS/DB/CS Versions ......................................................... 52  
Table 10. Base and Upgraded OS/DB/CS Versions ......................................................... 55  
Table 11. Base and Upgraded OS/DB/CS Versions ......................................................... 59  
Table 12. Base and Upgraded OS/DB/CS Versions ......................................................... 63  
Table 13. Base and Upgraded OS/DB/CS Versions ......................................................... 64  
Table 14. Base and Upgraded OS/DB/CS Versions ......................................................... 68  
Table 15. Base and Upgraded OS/DB/CS Versions ......................................................... 73  
Table 16. Base and Upgraded OS/DB/CS Versions ......................................................... 76  
Table 17. Migration Checklist ......................................................................................... 79  
Table 18. Premigration Tasks ........................................................................................ 80  
Table 19. DFS .NET Productivity Layer Upgrade Matrix .............................................. 99  
Table 20. Base and Upgraded OS/DB/CS Versions ......................................................... 105  
Table 21. DMCL API methods and corresponding DFC methods .................................. 112  
Table 22. New object types ......................................................................................... 119  
Table 23. Changed object types .................................................................................... 120  
Table 24. New object properties ..................................................................................... 127  
Table 25. Name changes for existing dfc.properties since version 6.5 and dfc.new properties ................................................................. 129  
Table 26. dfc.properties keys migrated from dmcl.ini file ........................................... 132  
Table 27. Obsolete session configuration options ......................................................... 134  
Table 28. Obsolete dfc.properties keys ......................................................................... 135
This guide describes the necessary steps to upgrade an existing EMC Documentum 6.5 SP2, 6.5 SP3, 6.6, 6.7 SPx, 7.0, 7.1 implementation to Documentum 7.x

**Intended audience**

This guide is for IT personnel who are upgrading the Documentum system, including Documentum custom applications.

**Document scope**

This guide describes how to upgrade a Documentum system and migrate customizations to the upgraded Content Server. Refer to the *EMC Documentum Content Server Installation Guide* for additional detailed planning information.

The release for upgrade to Documentum 7.x is supported from Documentum 6.5 SP2 onwards. If you are upgrading an earlier version, first upgrade to Documentum 6.5 SP2, and then to Documentum 7.x.

For migration, this guide provides appropriate instructions to move existing customizations in Documentum 6.6 to Documentum 7.x. This guide includes instructions to optimize your system by disabling unwanted new behaviors or activating earlier features that have been deprecated or “turned off” by default for 6.5 and later versions.

This guide does not focus on new features, except where a new feature changes or replaces existing behavior in custom applications.

All references to *6.5 and later* in this document refer to Documentum 6.5 SP2 and all versions that follow it, including Documentum 7.x.

All references to *7.x* in this document refer to Documentum 7.2.

**Revision history**

The following changes have been made to this document.
### Terminology

The following table provides a definition of the commonly used terms in this guide.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade</td>
<td>Refers to moving seamlessly from a previous version of the software to a more recent version. When hardware and third-party applications are compatible with the new version, and the existing version supports direct upgrade, an in-place move from an earlier version of EMC products can be performed.</td>
</tr>
<tr>
<td>Migration</td>
<td>Refers to moving customizations from one Content Server instance to another. It can refer to moving from an unsupported environment to a supported one, such as an upgrade that cannot be done in place due to lack of compatibility or the need to update/change hardware, or the need to move from an unsupported operating system/platform to a supported one. It can also refer to moving data from one location, server, or repository to another. The process of migration involves creating a repository and then copying the content from the old repository to the new repository.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Refers to software components that are intended to work together seamlessly. For example, different clients that can independently modify objects in the repository without conflicts, or an environment where Content Server applications, repositories, or client applications of different</td>
</tr>
</tbody>
</table>
versions coexist in an implementation (mixed version environments) without conflicts or errors.
Chapter 1

Upgrade and Migration Overview

This chapter provides a conceptual overview of upgrade and migration of a Documentum instance. Topics include:

• Upgrade and Migration, page 13
• Understanding Migration, page 14
• Order of New Product Installation, page 15
• Order of System Updates, page 16

Upgrade and Migration

This guide covers upgrade and migration of Documentum platform applications.

You must be on Documentum 6.5 SP2 or later to perform an in-place upgrade to Documentum 7.x. Direct in-place upgrade to Documentum 7.x is supported from Documentum 6.5 SP2, but only if the existing operating system/database/hardware combination is also supported for Documentum 7.x. Otherwise, a migration is required to move to a supported platform. The EMC Documentum Environment and System Requirements Guide provides information about the supported operating system, database, and hardware in Documentum 7.x.

Note: The EMC Documentum System Upgrade and Migration Guide that is included with your source and target versions provides instructions on upgrading from a version earlier to Documentum 6.5 SP2.

Migration refers to moving from an unsupported version to a supported version of the same operating system and database. It does not refer to moving from one operating system/database to another (for example, Solaris to Linux). Migration between operating system/database platforms requires a fresh installation and engagement with Professional Services.

You can migrate existing customizations such as DocApps, Documentum Archive (DAR) files, and business objects. Table 1, page 13 shows components that can be migrated, upgraded, or both.

Table 1. Upgrade and Migration of Product Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Migrate</th>
<th>Upgrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Server</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Custom DocApp/DAR files</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
If you are installing a new Content Server instance, move and modify (as required) the custom Java methods, DocApps, DARs, SBOs, and TBOs to the new Content Server instance.

Note:
- DocApps, SBOs, TBOs (BOF2 version), and Java methods bundled as SBOs continue to work in an upgraded Content Server.
- You can disable the new features or activate prior features that have been deprecated or turned off by default for Documentum 7.x. This guide does not describe new features, except where they change or replace existing behavior in custom applications.

Check the installation guide for each application that you are upgrading for specific considerations.

Before you perform the upgrade, check the list of products that are compatible with Documentum 7.x provided in the *EMC Documentum Environment and System Requirements Guide*.

The *EMC Documentum Content Server System Object Reference Guide* provides information about new, changed, deprecated, and obsolete object types and properties.

For information about deprecated, new, and obsolete properties in `dfc.properties`, see `dfcfull.properties`.

### Understanding Migration

Migration is a straightforward process. You document the current configuration, plan your upgrade configuration, and then upgrade the individual system components in a sequence that minimizes impact on your users.

Migration can be separated into two basic tasks:
- Install and configure Documentum 6.5 SP2 or later software.
- Move configurations and customizations to the new servers.
  - Enable features that you want to keep.
  - Disable new features that you do not want.
  - Enable new features for existing custom components.

The Chapter 5, Migrating Content Server chapter provide more information about the migration process.

Most of the new features in Documentum 6.5 SP2 and later versions are enabled by default. This guide provides the steps for enabling new features that are not enabled by default.
Order of New Product Installation

Figure 1, page 15 shows the recommended installation order for new Documentum systems. The server in this diagram is the host for the RDBMS, Content Server, or Index Server.

Figure 1. System Installation Order, New Documentum System

1. Install and configure network and server hardware
2. Install and configure network software
3. Install server and client OS
4. Install RDBMS
5. Install Documentum Server
6. Install Process Engine
7. Install custom DocApps or DARs
8. Set up integrated 3rd party applications (LDAP, SSO)
9. Install clients (DFC, DFS)

The recommended installation and upgrade order for new Documentum systems is as follows.

1. Documentum Content Server
2. Documentum Administrator (DA)
3. Documentum XML Store
4. Documentum Content Storage Services (CSS)
5. Documentum Trusted Content Services (TCS)
6. Documentum Content Services for Centera (CSEC)
7. Documentum Content Intelligent Services (CIS)
8. Documentum xPlore
9. Documentum High Volume Server (HVS)
10. Documentum Content Transformation Services (CTS)
11. Documentum Thumbnail Server
12. Documentum Branch Office Caching Services (BOCS)
13. Documentum Foundation Services (DFS), including any custom DFS applications
14. Documentum Content Management Interoperability Services (CMIS), including any custom CMIS applications
15. Documentum Foundation Classes (DFC), including any custom DFC applications

16. xCelerated Composition Platform (xCP) 2.x
   - Documentum Process Engine
   - Documentum Process Integrator

The coexistence of xCP 2.x and Documentum 6.7.x applications on the Documentum 7.x platform exists. The “Upgrading to enable xCP and Documentum clients to coexist” section in the EMC Documentum System Upgrade and Migration Guide, Release 6.7 SP2 provides information about the coexistence of xCP 2.x and Documentum 6.7.x applications scenarios.

**Order of System Updates**

Figure 2, page 16 shows the recommended order in which to upgrade system components. The server in this diagram is the host for the RDBMS, Content Server, or Index Server.

**Figure 2. System Update Order, Existing Documentum System**

![System Update Order Diagram](image)

⚠️ **Caution:** For Content Server, host operating system, or RDBMS upgrades, ensure that the product version is supported by the Content Server version you are installing. For application
server operating system or server upgrades, ensure that the product version is supported by the WDK-based application you are installing. The *EMC Documentum Environment and System Requirements Guide* covers this information.

In some cases, you must uninstall upgrades to existing Documentum system installations before installing a new version.

The *EMC Documentum Environment and System Requirements Guide* provides information on Documentum product compatibility. When there are version compatibility restrictions, upgrading one product generally requires upgrading interoperating products to the same version or to a major version family. In most cases, compatibility conflicts result from client applications that add new functionality to Content Server. In these cases, upgrading Content Server before the client application is especially important.
Chapter 2

Planning the Documentum System Upgrade

Upgrading a system requires planning. 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 Documentum 6.5 SP2, 6.5 SP3, 6.6, 6.7 SPx, 7.0, and 7.1 to Documentum 7.x.

Topics in this chapter include:

- System Upgrade Strategies, page 19
- Changes that Impact Content Server Upgrade or Migration, page 22
- Mapping Your Current Configuration, page 23
- Designing a Documentum 7.x Configuration, page 26
- Planning Upgrade and Migration to Documentum 7.x, page 28
- Planning Upgrade for Repositories in a Federation, page 30

System Upgrade Strategies

A Documentum system upgrade involves development, test, and production phases.

- Development — In this phase, you move customizations from an old product version to the new version and then verify that they still work properly.

- Test — In this phase, you deploy and run the full set of products to emulate the production system as closely as possible. This is frequently done on virtual hosts. Once all your system tests pass, you can upgrade the production system.

- Production — In this phase, you upgrade the existing production system in place with the verified customizations.

**Note:** The Content Server/database component (the repository) is the only part of the system for which there is an upgrade script. All other system product components require fresh installation.

The upgrade strategy provided in this section addresses upgrading all products in the system to the same version number, resulting in a homogeneous system.

**Figure 3, page 21** shows the high-level decision points involved when moving from a test system to a production system. Functional testing of new customizations and manual migration of existing customizations into new client version is a part of the development phase.
If you want to upgrade the repository, create a copy of the production repository in your test system on which you can run the upgrade. The Setting Up a Test Environment, page 28 section provides more information about creating a repository copy. If you want to change the database operating system, you can use the utilities available through the third-party database to export the database and import it into a new database instance on the different operating system. After running the Content Server configuration program to reestablish the connection between the existing Content Server instance and the new database instance, run Content Server to upgrade the entire repository.

If you are performing a fresh install instead of an upgrade, migrate your data files to the new Content Server and database instances. There are several third-party utilities, such as Crown Partners, Bluefish, and FME, which you can use to perform this data migration.

An EMC Documentum system requires a global registry repository that matches the version family of the system clients. The global registry is a central location used to store common objects used by all repositories, such as SBO network locations, BOCS settings, and user settings. After installing or upgrading the test system repository, install a global registry repository that matches the version of the client applications, and install the client software. If your client software versions are to remain as the same version as your production system, you can copy the customized files from your production system directly over to the same version client instance on the test system. If the client version software is different, migrate your customizations to the new client files.

After migrating to the test system, ensure that your system is running properly by conducting system tests. After all your system tests pass, you can upgrade the production system. Typically, your production system is taken offline for a weekend while performing an in-place upgrade.

**Note:** You can use virtual machine hosts for the entire system or for system components. Using virtual machines, you can swap out preupgraded system images on the same physical host to minimize the downtime of an in-place upgrade.

The production system contains new content and full-text indexes generated since the repository was copied or you migrated your data to the new repository.
Figure 3. System Upgrade Scenarios

- Fresh install or upgrade?
  - Fresh install
    - Export legacy data from production repository.
    - Install database software.
    - Install Content Server and create new repository.
    - Import legacy data.
  - Update Global Registry version corresponding to version of client applications on production system.
    - Install same version of client applications as on production systems.
    - Apply client application customizations from production system.
- Copy production repository to test environment.
- Change database OS?
  - Yes
    - Export database files.
    - Install database files into new database instance.
    - Install or upgrade to the latest client libraries (64-bit), as required.
    - Run Content Server configuration program.
  - No
    - Upgrade client applications?
      - No
        - Run systems tests.
      - Yes
        - Upgrade Global Registry version corresponding to new version of client applications.
        - Install new versions of client applications.
        - Apply client application customizations from production system.
Changes that Impact Content Server Upgrade or Migration

This section describes miscellaneous changes that can affect the migration to Content Server 7.x.

Linking to the 32-bit Oracle Client Library on Linux

In Documentum 6.7, for Linux, the way Content Server links to the Oracle Client library was changed. Prior to Documentum 6.7, a static link to the 32-bit Oracle Client library was used. From 6.7 onwards, Content Server does not link with the 32-bit Oracle Client library at build time, but rather dynamically links to it at runtime. Consequently, you must install a 32-bit Oracle Client on the Content Server host, and add the directory containing the client libraries to the appropriate environment variable (LD_LIBRARY_PATH for Linux) of the repository owner before configuring or starting Content Server.

In Documentum 7.1, all the UNIX ports of Content Server are 64-bit. Therefore, the 32-bit Oracle client libraries are not required and the LD_LIBRARY_PATH environment variable should directly be set to the 64-bit Oracle client libraries.

If you are upgrading from a Content Server version prior to Documentum 6.5 SP2, the earlier version of the repository does not start when Oracle version is upgraded to Oracle 11g Release 2. You need to upgrade Content Server to Documentum 6.7 to perform any operations with the repository before upgrade.
If you are upgrading from Documentum 6.5 SP2 or 6.5 SP3 or 6.6 or 6.7 or 7.0 to Documentum 7.x and want to perform any operations with the repository after the Oracle version upgrade to Oracle 11g Release 2, the *EMC Documentum Content Server Installation Guide* for instructions on configuring the repository with Oracle.

**Upgrading Multiple Repositories to the Same Version**

When upgrading Content Server from 6.5 SP2 or later to 7.x, first upgrade the software and then the repository.

If Content Server comprises of multiple repositories, ensure that you upgrade all repositories to 7.x. You cannot have multiple repositories of different versions on the same Windows host since different method server binaries are used for different versions. If you retain a repository at a 6.5 SP2 version, the 6.5 SP2 repository uses the latest 7.x method server executable to launch JMS based on the registry entries on the Windows host.

It is recommended that you upgrade all repositories in Content Server to the same version.

**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 may already have similar diagrams from which you can get much of this information. If you do not, be sure to keep a copy of the existing plan to help with future migrations.

Take the time to verify that any existing diagrams reflect the current configuration.

Complete one copy of Table 2, page 23 for each server host and client configuration used in your current system (for example, Content Server, full-text indexing server, Federated Search Server, or application server).

**Table 2. Content Server and Database Server Host Worksheet**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and Processors</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>Operating system and version</td>
<td></td>
</tr>
<tr>
<td>Content Server version</td>
<td></td>
</tr>
<tr>
<td>RDBMS and version</td>
<td></td>
</tr>
<tr>
<td>Repository size</td>
<td>Number of objects:</td>
</tr>
<tr>
<td></td>
<td>Storage space required:</td>
</tr>
</tbody>
</table>
Global Registry? [ ] Yes [ ] No
Java/JRE version
DFC version
Other product version
Other product version
Other product version

Table 3. Application Server Host Worksheet

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and processors</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>Operating system and version</td>
<td></td>
</tr>
<tr>
<td>HTTP Server version</td>
<td></td>
</tr>
<tr>
<td>Java version</td>
<td></td>
</tr>
<tr>
<td>DFC version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Index Server Host Worksheet

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and processors</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>Operating system and version</td>
<td></td>
</tr>
<tr>
<td>HTTP server version</td>
<td></td>
</tr>
<tr>
<td>Java version</td>
<td></td>
</tr>
<tr>
<td>DFC version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
</tbody>
</table>
### Table 5. Client Machine Worksheet

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system and version</td>
<td></td>
</tr>
<tr>
<td>Browser and version</td>
<td></td>
</tr>
<tr>
<td>Java version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
<tr>
<td>Other product and version</td>
<td></td>
</tr>
</tbody>
</table>

### Table 6. Customized Components Worksheet

<table>
<thead>
<tr>
<th>Product</th>
<th>Customized Components</th>
<th>Customization type</th>
<th>Customization Description</th>
<th>Disposition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.5 and later Compatible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Needs changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Obsolete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.5 and later Compatible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Needs changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Obsolete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.5 and later Compatible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Needs changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Obsolete</td>
</tr>
</tbody>
</table>
Designing a Documentum 7.x Configuration

This section discusses some of the design decisions you must make before implementing a Documentum 7.x configuration. Departmental systems are configurations where 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 dispersed 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. When upgrading to Documentum 7.x, ensure that at least 10 GB of memory is available.

You can also make an estimation on the required memory by calculating the memory consumption of each of the Documentum executables. The following table lists the memory consumption for each executable of Documentum based on their count.

<table>
<thead>
<tr>
<th>Documentum executable name</th>
<th>Count</th>
<th>Memory consumption (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMBASIC method server (Master)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Agent</td>
<td>5</td>
<td>2560</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Each agent requires 512 MB of memory.</td>
</tr>
<tr>
<td>Java Method Server</td>
<td>1</td>
<td>1300</td>
</tr>
<tr>
<td>Agent executable</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Documentum.exe</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>9</td>
<td><strong>4.373 GB</strong></td>
</tr>
</tbody>
</table>

**Note:** The table lists the memory consumption for a single docbase if the database is installed in another machine. If you have multiple docbases, multiply the total memory with the number of docbases. For example, if you have two docbases, the total memory required would be 8.6 GB (that is, 2 X 4.3). The memory consumptions values are derived or obtained after testing the product in the EMC testing environment. Every effort is made to simulate common customer usage scenarios but actual results may vary due to differences in hardware and software configurations, data, and other variables.
Upgrading Third-party Software

Verify that the third-party software, such as operating system, database, and so on, you are currently using with the existing version is still supported, or upgrade to the supported versions as necessary. If the third-party component does not have a direct upgrade path to the supported version, then there is no direct upgrade path for the Content Server upgrade. The *EMC Documentum Environment and System Requirements Guide* provides information about supported third-party software version.

Planning for Global Registries

Designate one of the repositories in your version 6.5 or later system as the *global registry*. Decide which of your repositories to use as the global registry. If you already have a Documentum 5.3 SP6, 6.0, or 6.0 SPx global registry, upgrade to Documentum 7.x.

During repository configuration, you are prompted with the message **Do you want to add this repository to another global repository**. You can select one of the following options:

- **Yes**
  
  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.

- **No**
  
  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; 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.

Regardless of whether you designate the repository as a global registry or not, the global registry user is created for all repositories. The global registry user (dm_bof_registry), is the repository user whose account DFC clients use to connect to the repository to access required service-based objects and user information. The user has read access to objects in /System/Modules only.

- **If you configure the repository as a global registry**, provide the user login name and password for the user and set the user state to *Active*.
  
  This can be any arbitrary user login name and password. Do not use the credentials of the repository or installation owners.

- **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. The *EMC Documentum Content Server Installation Guide* provides instructions on enabling the repository as a global registry.
Mapping a Documentum 7.x 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 23).

Planning Upgrade and Migration to Documentum 7.x

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 Documentum 6.5 or later system. The migration paths described in this section allow your applications to continue working and minimize impact on your users, but your users cannot get the full benefits of features in Documentum 6.5 or later versions 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. Set up an environment that includes the same hardware, RDBMS, and software configurations as your production system, including a copy of your production repository. Setting up a test environment allows you to practice migrating your systems, as well as troubleshoot any migration problems, before committing changes to your production system.

Creating the Test Environment

You cannot create copies of more than one repository in a single new installation if the repositories were created in different installations.

Use the instructions for creating a repository copy on the same platform as the original repository. The procedure is not supported for moving a repository from one platform to another.

Before upgrading a repository, create an environment in which to test the upgrade process. To do this, create a new installation using the original Content Server software version, copy the repository, copy the content files, and upgrade that copy. Perform tests on the copy ensuring to exercise standard functionality and customizations. After the upgraded copy is tested completely, upgrade the original repository.

For example, if you are copying two repositories, Paris and London, that were created in separate Content Server installations, you need to copy them to separate Content Server installations. Creating a repository copy requires you to copy the aek.key and dbpasswd.txt files from the original repository host to the repository copy host, because each repository copy must have access to the aek.key and dbpasswd.txt files from its original installation.

If you are copying two repositories, Tokyo and Beijing, that were created in the same Content Server installation, you can create their copies in the same new installation, with the aek.key and
**dbpasswd.txt** files from the original installation copied to the installation where you create the copies.

The Method 2: Copying a Repository, page 83 section in the Chapter 5, Migrating Content Server chapter provides the steps for copying a repository.

---

**Migrating a TCS-Enabled Docbase to a New Server**

To migrate content from an encrypted store in a source repository to a target repository, perform the following steps:

1. Copy the content from an encrypted store to a non-encrypted store on the source repository.
2. Copy the repository. The Method 2: Copying a Repository, page 83 section in the Chapter 5, Migrating Content Server chapter provides the steps for copying a repository.
3. Create a new encrypted store in the target repository and migrate the content from the non-encrypted store to the newly created encrypted store in the target repository.

---

**Client-First Migration**

If your system uses only 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** — Parallel streaming from Accelerated Content Server (ACS) is used only if both ACS and Unified Client Facilities (UCF) (WDK or DFS applications) are version 6.5 or later. Parallel streaming from Branch Office Caching Services (BOCS) is used only if ACS, BOCS, and UCF are version 6.5 or later.

**Note:** When you upgrade the BOCS to version 6.5 or later, 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 to 6.7, specify the version as 2.1. For BOCS 7.0, specify the version as 2.3.

**WDK Clients Compatibility and Migration** — WDK-based clients are compatible with Documentum 5.3 SP6 and 6.0 Content Servers. Based on the version to which you have migrated, some WDK features of that version are available, such as features that run in JavaScript on the client. Other features are not available until you complete the migration to Documentum 6.5 or later, such as lightweight system objects, data partitioning, batch processing, and scoping.

The following procedures describe the steps to migrate from Documentum 5.3 SP6, 6.0, or 6.0 SPx to Documentum 6.5 or later, migrating the clients first.

**To migrate from Documentum 5.3 SP6, 6.0, or 6.0 SPx to Documentum 7.x:**

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. The EMC Documentum Content Server Installation Guide provides instructions for enabling a global repository. Documentum 6.5 or later clients with a Documentum 5.3 SP6 global registry are not supported.
3. Upgrade the Content Servers in place.
4. Restart Content Server.
5. Configure one Content Server as the version 6.5 or later global registry. If you had a global registry in version 6.0, you can upgrade that server in place using the same settings.

Planning Upgrade for Repositories in a Federation

A Federation is two or more repositories that are bound together to facilitate management of global users, groups, and access control lists (ACLs) in a multi-repository distributed configuration. One repository in the set is the governing repository. The remaining repositories are member repositories.

Keeping objects synchronized in multiple repositories can be time consuming and error-prone when the work is done manually in each repository. A repository federation automates much of the process.

Only certain combinations of different repository versions can work together as a Federation. Plan your upgrade so that all participating repositories are supported.

When you upgrade repositories (that work together as a Federation), ensure that the upgraded repository versions (in a nonhomogeneous Federation) can work together as a Federation.

The EMC Documentum Content Server Distributed Configuration Guide provides more information about configuration requirements in a Federation.

Guidelines for Upgrading a Distributed Configuration

Use these guidelines when deciding how to upgrade a distributed configuration.

Repository Federations — EMC Documentum supports federations that contain repositories of different versions. In a mixed version environment, upgrade the governing repository first.

Repositories with Object Replication — Upgrade the source repository, then the target repositories. If you have a group of repositories where each repository is both a source and a target, then the upgrade can begin with any of the repositories. This situation can apply, for example, if objects are replicated from repository A to repository B, repository B to repository C, and from repository C to repository A. Although you can replicate between repositories that are different versions, attributes that are only in the newer version cannot be replicated.

Repositories with Distributed or Load-Balanced Content — Shut down the primary Content Server and all remote Content Servers. Upgrade the primary Content Server first, then upgrade the remote Content Servers.

The Upgrading Content Server in a Distributed or Load-Balanced Configuration, page 45 section provides the steps for upgrading Content Server in a distributed or load-balanced configuration.
Planning the Upgrade or Migration of the AEK Key to Lockbox

From the 7.2 release, you can upgrade the AEK key to a stronger algorithm during the repository upgrade as follows:

• By choosing stronger algorithm for the AEK key: You can upgrade the AEK key algorithm to AES 128/192/256 algorithm. The Post-Migration Tasks section of Migrating Content Server 6.5 SP2 to 7.2 — Windows/SQL Server, page 105 contain the procedure for this process.

• By enabling lockbox for the AEK key: You can move an existing AEK key to lockbox or create a new upgraded AEK key in the lockbox. The Post-Migration Tasks section of Migrating Content Server 6.5 SP2 to 7.2 — Windows/SQL Server, page 105 contain the procedure for this process.

Note: During the upgrade process, use the same passphrase for the new AEK key. After the upgrade is complete, change the passphrase. Use `dm_crypto_change_passphrase` to change the passphrase and then run the `dm_crypto_boot` utility with the new passphrase.

For example:
```
dm_crypto_change_passphrase -keyname CSaek [-lockbox lockbox.lb -lockboxpassphrase <lockboxpassphrase>] -passphrase genuine -newpassphrase glorious
```

Caution: If you select the Upgrade AEK key option, it results in changing the AEK key and re-encrypting the repository keys and other related data such as CNT files. Hence, it is recommended to take a backup of the AEK key and CNT files before the upgrading the AEK key.

The EMC Documentum Content Server Installation Guide contains detailed information about creating strong AEK/lockbox options.

Enabling Remote Key Management

From Release 7.2 onwards, you can enable Remote Key Management (RKM) for supported upgrades during upgrade or post upgrade.

Once you enable RKM, the Docbase Key (DBK), Login Ticket Key (LTK), File Store Key (FSK), and Private/Public Key (PPK) retain the same value as that before upgrade, even if you upgrade the AEK key algorithm. In the RKM server, the key classes need to be created with the same algorithm that was used for these keys before upgrade. Only the value of the AEK key is changed.

For migration, it is recommended that you enable remote key management after migration.

The EMC Documentum Content Server Installation Guide contains detailed information about configuring the remote key management.
Chapter 3

Upgrading Content Server

This chapter describes how to upgrade from a previous release and how to upgrade repositories to Content Server 7.x. Always consider upgrading one Documentum product within the context of upgrading the entire Documentum system.

Note: You cannot upgrade to Content Server 7.x from a version earlier than 6.5 SP2. If your current installation is an earlier version, upgrade it to Content Server 6.5 SP2 or later before you upgrade to 7.x.

This chapter contains the following topics:

- Upgrade Checklist, page 33
- Upgrading to 64-bit Content Server, page 37
- Upgrading the Content Server Software, page 40
- Upgrading Content Server Installed with Cluster Services, page 44
- Upgrading Content Server in a Distributed or Load-Balanced Configuration, page 45
- Post-Upgrade Tasks, page 48

Each step in the upgrade process must be to a platform that is fully supported by Documentum. Depending on the Content Server release from which you are upgrading, you may need to upgrade the operating system or database. The documentation provided by the operating system or database vendor contains information on upgrading those components of the system. After each upgrade step, test the repository to ensure that all functions are normal.

Note: Windows Server 2003 is not a supported environment for Content Server 7.x. You must upgrade your operating system to a supported environment before upgrading Content Server.

Caution: After upgrading, you cannot revert to previous versions of Content Server.

Upgrade Checklist

Perform the following tasks for upgrading Content Server:

1. Review the EMC Documentum Content Server Release Notes.
2. Review the EMC Documentum Content Server Installation Guide.
3. If you are installing the xPlore indexing server, review the *EMC Documentum xPlore Installation Guide*.

4. Review the Upgrading to 64-bit Content Server, page 37 section on what sequence to use in upgrading your installation, especially if you are upgrading from a 32-bit to a 64-bit Content Server.

5. Back up the repository. For the steps, refer to the *EMC Documentum Content Server Administration and Configuration Guide*.

6. Optionally, you can take a backup of all users that are part of the *Admin group* and any customized attribute like *group_address*.

7. Decide whether to enable extended services, such as:
   - Retention Policy Services
   - Trusted Content Services
   - Content Services for EMC Centera
   - Content Storage Services
   - Records Manager

   The “Content Server optional modules” section in the *EMC Documentum Content Server Installation Guide* provides more information.

8. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. Refer to the database documentation for the steps.

9. Ensure that you have sufficient disk space on the computer hosting the database.

10. Run the repository consistency checker script and correct any errors you find. The *EMC Documentum Administrator User Guide* and Chapter 3, Upgrading Content Server provides the steps for running the consistency checker.

11. Ensure that the *dm_server_config* object is unlocked. The Ensuring that the *dm_server_config Object is Unlocked*, page 37 provides more information.

12. Shut down the repository and all servers running against the repository.

13. Close the Documentum Server Manager User Interface.

14. Shut down any local connection brokers.
15. On Linux/Solaris/AIX:
   a. Set the `$DOCUMENTUM` environment variable same as that in the base version.
   b. Modify the `$DM_HOME` variable in the installation owner’s .cshrc or .profile file to point to `$DOCUMENTUM/product/7.x`. The *EMC Documentum Content Server Installation Guide* provides more information.
   c. Set the `$DOCUMENTUM_SHARED` environment variable same as that in the base version. Do not delete this environment variable.
   d. Modify the library path variable in the installation owner’s .cshrc or .profile file to point to the location of the shared libraries required by the server. The *EMC Documentum Content Server Installation Guide* provides more information.
   e. If you are using Oracle as the database, modify the `$ORACLE_HOME` environment variable to point to the 64-bit libraries.
   f. Determine the root password. This is the operating system root password. The root password is required to complete the upgrade. Refer to the Linux/Solaris/AIX documentation for more information.

16. Determine the installation owner username and password. Refer to the *EMC Documentum Content Server Installation Guide*, and consult the database administrator.

17. Determine the names of the repositories you are upgrading.

18. Determine the Content Server version from which you are upgrading.

   **Note:** In a fresh installation of Documentum 7.x, all files are installed in a single directory as defined by the `%DOCUMENTUM%` environment variable. For example, in a fresh installation, the default directory is `C:\Documentum`. In the earlier versions of Documentum, the default directory was `C:\Program Files\Documentum`. When you upgrade the older versions of Documentum to 7.x, the existing installation directories are retained. Only the new 64-bit Java files are installed in the new Documentum 7.x installation directory.

---

**Changing the Database Operating System and Version**

When migrating your database to a new operating system (host) and database version, complete the database migration first before upgrading Content Server. After migrating the database, run the Content Server configuration program to reestablish the repository with the new database instance. Then upgrade Content Server to upgrade the entire repository.

Refer to the database vendor documentation for information on migrating the 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.

   **Note:** While upgrading the Windows operating system, the `\etc\services` file is replaced as a part of the upgrade. Because of this, the entries that were added for the repository before the upgrade is lost. After upgrading the operating system, manually add the repository service entries to the `\etc\services` file.
Preparing DB2 databases for Upgrade

Prior to upgrading Content Server on AIX with a DB2 database, create a temporary tablespace with an 8K page size. Then, backup all tables into the new tablespace.

To create an 8K temporary tablespace, run the following command:

db2 CREATE TEMPORARY TABLESPACE TEMPSPACE2 PAGESIZE = 8192

Migrating the Database to UTF-8

If the database was installed with a code page other than UTF-8 under a previous version of Content Server, you do not have to migrate the database to UTF-8 to upgrade Content Server. However, to use the multilingual functions of Content Server, migrate the database to UTF-8.

EMC Documentum supports upgrading repositories by using the existing database code page.

- On Oracle, you can migrate existing repositories to Unicode using the tools provided by Oracle. Contact Oracle for any support you require in migrating the database.

- On DB2, all repositories should have been created by using Unicode. They do not need to be migrated. If the DB2 repository does not use Unicode, migrate the database by using IBM tools. Contact IBM for any support you require.

- On Microsoft SQL Server, you cannot migrate the database to Unicode.

The database_refresh_interval Key

During Content Server installation or upgrade, the change checker process runs once per minute by default. The process updates type caches as types are created or altered. Before you upgrade, ensure that the key is set to 1 minute or delete it from the server.ini file.

Ensuring the Completion of Automatic Tasks Before Upgrading

If you are using a repository of Documentum 6.5 or earlier, ensure that all automatic tasks are completed before shutting down the repository for upgrade; otherwise, unfinished automatic tasks will fail.

Use the following Documentum Query Language (DQL) query to obtain the number of active automatic tasks in the repository:

```sql
select count(r_object_id) from dmi_workitem where
r_auto_method_id> '0000000000000000' and
r_runtime_state in (0,1)
```

If the query returns a nonzero value, active automatic tasks still must be processed and you must wait for them to complete. If it returns 0, the repository contains no more active automatic tasks, and you can safely stop the repository. If the query returns 0, run the query a few more times to ensure that no new automatic tasks are being generated.
Ensuring that the dm_server_config Object is Unlocked

If you attempt to upgrade Content Server and the dm_server_config object is locked, the upgrade may fail. To check if the object is locked, log in to your database as the database owner and use the following SQL query to get the object ID of the server configuration object:

```
SQL> select r_object_id from dm_server_config_s
```

Use the object ID in the following query to verify whether the configuration is locked:

```
SQL> select r_object_id, r_lock_owner from dm_sysobject_s
where r_object_id = '<object ID>'
```

If there is a lock owner, then the object is locked.

To unlock the object, use the following SQL (except for Oracle):

```
SQL> update dm_sysobject_s set r_lock_owner = '' set r_lock_machine = ''
set r_lock_date = ' ' where r_object_id = '<object ID>'
```

For Oracle, use:

```
Oracle> update dm_sysobject_s set r_lock_owner = '' set r_lock_machine = ''
set r_lock_date = null where r_object_id = '<object ID>'
```

Commit the change:

```
SQL> commit
```

Finally, restart the repository.

Upgrading to 64-bit Content Server

Upgrading to the 64-bit Content Server is supported only if there is an upgrade path on the underlying operating system and RDBMS. For example, upgrading the 32-bit Content Server on 32-bit Windows Server 2008 to the 64-bit Content Server is not supported because there is no supported upgrade path from 32-bit Windows Server 2008 to 64-bit Windows Server 2008.

If your operating system and database meet the requirements specified in the *EMC Documentum Environment and System Requirements Guide*, you can directly upgrade from 32-bit Content Server 6.5 SP2 or later directly to 64-bit Content Server 7.x. For this upgrade path, the underlying operating system must be 64-bit.

Use the following approach to upgrade from 32-bit Content Server 6.5 SP2 or later to 64-bit Content Server 7.x:

1. Upgrade the operating system to the supported version, if necessary.
2. Upgrade the database, if necessary.
3. Perform one of the following steps depending on the type of database being used:
   - **SQL Server**: Upgrade or install a 64-bit version of the database client on the Content Server host machine. When you install the 64-bit database client, copy the DSNs from the 32-bit ODBC driver to the 64-bit driver if used by your database. When you redefine the DSN,
use the same level or later level of client library. For more information, see 64-bit ODBC and DSN Libraries, page 38.

- **Oracle**: Create an ORACLE_HOME environment variable in Windows that points to the location of the 64-bit TNSNAMES.ORA file. Copy the entries from the 32-bit TNSNAMES.ORA file into the 64-bit TNSNAMES.ORA file.

4. **Upgrade from 32-bit Content Server 6.5 SP2 or later to 64-bit Content Server 7.x.**

You may see database connection errors in the repository logs, since the combination of a 32-bit Content Server and a 64-bit database client is not supported. These errors can be ignored.

During the upgrade from 32-bit to 64-bit, Content Server, you cannot upgrade the authentication plug-ins that you have installed. You must replace the 32-bit authentication plug-ins with the 64-bit plug-ins. You can find the plug-ins in the \%DM_HOME\%\install\external_apps\authplugins folder.

- Do not probe log files before the entire upgrade is completed. Partial upgrade is not supported. Upgrade the binaries, the connection broker, and the repositories at the same time.

- For operating systems that do not support IPv6 in versions prior to Documentum 6.5, upgrade the operating system first. For example, in the case of Red Hat Linux, you must upgrade the operating system first if it did not support IPv6 in versions prior to Documentum 6.5.

- If the 32-bit Content Server is installed on a 32-bit operating system, migrate the repository to the 64-bit version of the operating system and then upgrade the Content Server. If the 64-bit version of the operating system is not supported, you must upgrade the operating system to the supported version before upgrading the Content Server.

- If you are migrating your database to a new operating system and database version, you must perform the migration before upgrading Content Server. After completing the database migration, run the Content Server configuration program to reestablish the repository with the new database instance; then upgrade Content Server to upgrade the entire repository.

Figure 4. Upgrade Steps from 32-bit Content Server to 64-bit Content Server

64-bit ODBC and DSN Libraries

64-bit Content Server requires 64-bit database client libraries for the Oracle database or Microsoft SQL Server. For the Oracle database, update the ORACLE_HOME variable to the 64-bit installed path, and tnsnames.ora should take the appropriate entries. For Microsoft SQL Server, update the DSN entry from 32-bit to 64-bit and verify that DSN is pointing to the 64-bit SQL libraries.
To migrate the ODBC and DSN libraries from versions for 32-bit Content Server to versions for the 64-bit Content Server, follow these steps:

1. Run the 32-bit ODBC DSN utility. Browse to C:\Windows\SysWOW64 and double-click odbcad32.exe. The ODBC Data Source Administrator dialog box opens.
2. Note down all the DSN entries. These are the 32-bit DSNs present in 32-bit operating system registry.
3. Run the 64-bit ODBC DSN utility. Browse to C:\Windows\System32 and double-click odbcad32.exe.
4. In the ODBC Data Source Administrator dialog box, on the System DSN tab, add the 32-bit DSN entries ensuring that the same values are used as in step 2 and click OK.

**Errors During Upgrade**

If you upgrade from 32-bit Content Server 6.5 SP2 or later to 64-bit Content Server 7.x using the second approach described in Upgrading to 64-bit Content Server, page 37, you might encounter errors. During the upgrade, at an intermediate stage where, for example, the 64-bit database client libraries are installed with 32-bit Content Server and a 64-bit RDBMS is running, the system will be in an unstable state. You can expect to see errors if any of these systems are running. Validation of features or functionality during this intermediate step is not permitted. Continue with the upgrade and run the 64-bit Content Server installer before testing the system.

For example, if you are migrating from 6.5 SP2 to 64-bit Content Server 6.7, you might see docbasic connection errors during the intermediate step of installing 32-bit Content Server 6.7. These errors are expected.

When upgrading a repository from 32-bit to 64-bit, Content Server automatically recompiles the docbasic expressions during their execution. This recompilation occurs because the underlying library changes from 32-bit to 64-bit.

This recompilation process can increase the execution time of the method, which contains these expressions. If a very small method timeout value was specified, the operation can result in a METHOD_TIMEOUT error. However, the method continues running even after the timeout error is reported.

Therefore, when upgrading a repository, watch out for these timeout errors and ensure the proper state of the method before retrying the operation.

**Components Not Supported after Upgrade**

FAST was the default search engine prior to the Documentum 6.6 Content Server. FAST is not supported on the 64-bit Content Server. At a minimum, disable the FAST component on the 64-bit Content Server. It is recommended that you uninstall the FAST component during migration from 32-bit Content Server 6.7 to 64-bit Content Server 6.7. Documentum 7.x uses the xPlore search engine. The EMC Documentum xPlore Installation Guide provides information about migrating FAST data to xPlore.
Migrating Custom Plug-ins

Migrate all 32-bit custom plug-ins to the 64-bit architecture. The 64-bit Content Server does not support 32-bit custom plug-ins.

Upgrading Content Server on Red Hat Linux 5.x Systems

Red Hat Linux does not support upgrading the operating system from Linux 5.x to 6.x. If you want to upgrade Content Server 6.7 or earlier running on a Red Hat Linux 5.x system to Content Server 7.x that runs on Red Hat Linux 6.x, you must follow the migration procedure described in Chapter 5, Migrating Content Server.

Upgrading the Content Server Software

The length of time required to upgrade a repository depends on the size of the repository. Allow sufficient time for backing up the repository and performing the upgrade.

To upgrade the Content Server software:

1. Back up the repository. Several third-party tools are available that you can use.
2. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
   
   Repository formats are upgraded by the dm_apply_formats.ebs script, which reads values from the formats.csv file. If the attributes of a format in the repository do not match the format descriptions in the formats.csv file, the script overwrites the existing values with the values in the file.
3. Run the Consistency Checker tool.
   
   Consistency Checker is a script that looks for repository inconsistencies, such as users with nonexistent groups, permissions sets with nonexistent users, and sysobjects that point to nonexistent content files. Fixing inconsistencies in the repository improves the quality of the data in the repository and results in a smoother upgrade. The syntax is:

   \[ \text{dmbasic} \ -f\text{consistency_checker.ebs} \ -e\text{Entry_Point} \ -- \text{repository_name} \ superuser \ password \]

   • \text{repository_name} is the name of the repository against which you are running the Consistency Checker tool.
   • \text{superuser} is the username of a repository superuser.
   • \text{password} is the password for the superuser’s account.

   The results of the Consistency Checker tool are directed to standard output.
4. Fix the inconsistencies reported by the Consistency Checker tool as errors.
   
   The EMC Documentum Content Server Administration and Configuration Guide provides information about the Consistency Checker tool.
5. Upgrade the operating system if necessary.
6. Upgrade the database if necessary.
7. Disable all jobs.
   - On Windows, disable jobs in all repositories on the host.
   - On Linux/Solaris/AIX, disable jobs in all repositories in the installation you are upgrading.
   **Note:** Ensure that you make a note of the arguments and values of the standard jobs especially of the LDAP Synchronization job, before the upgrade. After the upgrade, these jobs will have the default values and arguments. Hence it is recommended to save the settings before the upgrade.
8. For the upgrade on a Windows host, shut down the repositories and connection brokers.
   a. Click **Start > Programs > Documentum > Server Manager**.
   b. Select the correct Content Server and click **Stop**.
   c. Click the **Connection Broker** tab.
   d. Select each connection broker.
   e. Click **Stop**.
9. For the upgrade on a Linux/Solaris/AIX host, shut down the repositories and connection brokers.
   a. For each repository, run the dm_shutdown_repository script, where repository is the name of the Content Server to be stopped.
   b. Stop each connection broker using the dm_stop_docbroker utility.
      The *EMC Documentum Content Server Administration and Configuration Guide* provides the steps for using the dm_stop_docbroker utility.
10. Shut down the application server.
    - To shut down the application server on Windows, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
    - To shut down the Java Method Server service on Linux/Solaris/AIX, run script 
        \$DOCUMENTUM_SHARED/<jboss folder>/server/stopMethodServer.sh.
11. Run the Content Server installation program.
    a. To launch the 64-bit Content Server Installer:
       - On Windows, run serverSetup.exe.
       - On Linux/Solaris/AIX, run serverSetup.bin.
    b. The installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed. Click **Yes**.
    c. Accept the license agreement and click **Next**.
    d. Perform one of the following steps:
       - In the Windows installer, type the **installation owner password** and click **Next**.
       - In the Linux/Solaris/AIX installer, type the **Root User Password** and click **Next**.
    e. Review the installation summary and click **Install** to begin installation.
f. Specify if you want to enter license keys for optional modules of Content Server and click Next.
   • Yes: On the next page, select the optional modules you want to install and enter corresponding license keys.
   • No: You can always enter license keys for optional modules later using the Content Server configuration program.

g. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

   **Note:** When upgrading, if you select this option or the Configure later option, you will be prompted to select the connection modes (Native, Secure, and Native and Secure) for the repository upgrade.

   During the connection broker upgrade, you will not be prompted to select the connection modes.

12. Upgrade the connection broker.

a. If you want to manually enable the use of certificates when upgrading the connection broker, follow these steps:
   i. Stop the connection broker service.
   ii. Modify `broker.ini` file by adding following properties:
      • `secure_connect_mode`
      • `keystore_file`
      • `keystore_pwd_file`
      • `cipherlist`
   iii. Modify the `dfc.properties` file by adding following properties:
      • `dfc.security.ssl.truststore`
      • `dfc.security.ssl.truststore_password`
      • `dfc.security.ssl.use_existing_truststore`
   iv. Restart the connection broker service.

   The *EMC Documentum Content Server Installation Guide* provides more information about the `broker.ini` and `dfc.properties` properties and the values you can specify with examples.

b. In the Documentum Content Server Configuration Program wizard, select **Connection broker** and click Next.

c. In the Windows installer, type the **installation owner password** and click Next.

d. Select **Upgrade a connection broker**.

e. Select the connection broker to upgrade from the list, and click Next.

f. Complete the configuration, select **Perform additional configuration**, and click Next.

13. Upgrade the repository.

a. Click **Upgrade an existing repository**.
b. Select the repository to upgrade from the list, and click **Next**.

c. Type the **Connection Broker Port** and **Connection Broker Host** and click **Next**.

d. Select the **Connection Mode** for the repository and click **Next**. If you select the **Secure** or **Native and Secure** options, select **Use certificate** on the next page, provide the required details, and then click **Next**.

e. In the Windows installer, type the **SMTP server name** and the **owner’s email address** and click **Next**.

f. Select the module you want to activate for the repository and click **Next**.

g. Specify whether you want to change the xDB superuser password and click **Next**.

**Note:** This configuration screen only appears if you are upgrading from pre-7.0 Content Server to 7.x.

h. Type the **repository owner password** and the **Database administrator name** and **password**, and then click **Next**.

**Note:** During the upgrade process, a message might appear stating that the repository is not available. Before you click **Yes** in the message box, ensure that the repository is up and running by checking the `<Documentum_Home>\dba\log\<repository>.log` file.

i. After the upgrade completes, select **Finish configuration** and click **Next**.

14. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, restart Content Server again after you complete the Content Server configuration.

15. After you complete the Content Server configuration, create a nonunique index on the `dm_sysobject.r_object_id` and `r_aspect_name` properties by using the following `MAKE_INDEX` command:

```
EXECUTE make_index WITH type_name='dm_sysobject',
attribute='r_aspect_name',use_id_col=true
```

The inclusion of the `use_id_col` argument set to true automatically causes Content Server to include the `r_object_id` column in the index.

**Note:** Before you create the index, verify if it exists already. Check if `dm_sysobject` has an index on `r_aspect_name` and `r_object_id`. If the index does not exist, then create it by using the above DQL.

16. If you are upgrading a repository in a distributed environment that uses a BOCS and asynchronous write jobs, create an index on the `dmr_content_s(i_parked_state, r_object_id)` properties. Use the following `MAKE_INDEX` command to do so:

```
EXECUTE make_index WITH type_name='dmr_content',
attribute='i_parked_state',use_id_col=true,id_in_front=false
```

17. After the upgrade is complete: on Windows, delete directory `%DOCUMENTUM%<OLD_JBOSS_HOME>` (for example, `c:\documentum\jboss4.3.0`); On Linux/AIX/Solaris, delete directory `$DOCUMENTUM_SHARED/<OLD_JBOSS_HOME>`.

18. Enable all the disabled jobs.

a. In Documentum Administrator, navigate to **Administrator > Job Management > Jobs**.

b. For each of the previously disabled jobs, right-click the job and select **Properties**.

c. In the Properties window, set the **State** option to **Active**.
**Note:** After the upgrade is complete, the standard jobs will have the default values and arguments. Hence set the values of the jobs again to the custom values and arguments assigned before upgrade.

19. On a Windows upgrade, the **Startup Type** is set to **Manual** for the Documentum Docbase Service **repository name** service. If you want the repository to automatically start after a server reboot, navigate to **Start > All Programs > Administrative Tools > Services**, and set the **Startup Type** to **Automatic**.

20. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete the upgrade.

21. Optionally, run `dm_filestore_unique.class` in `%DM_HOME%/install/tools` (`$DM_HOME/install/tools` in Linux/Solaris/AIX) to create a filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

22. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure that you manually update the backed up customized attribute.

**Note:** If you are upgrading Content Server in a cluster environment and are using a non-default datapath for Content Server, update `headstart.ebs` to retrieve the correct location object: `retrieve,c, dm_location where file_system_path like '%content_storage_01%'`

### Upgrading Content Server Installed with Cluster Services

Use the following procedure to upgrade Content Server installed with Microsoft Cluster Services. This procedure applies to upgrades on active/passive, active/active, single-repository, and multirepository configurations.

**To upgrade an active/passive, single-repository cluster:**

1. Shut down the Content Servers on both nodes.  
   This shuts down the repository.
2. Shut down both hosts.
3. Restart the first node.  
   Do not restart Content Server on the first node.
4. On the first node, upgrade the Content Server software. The Upgrading the Content Server Software, page 40 section provides the steps.
5. Upgrade and configure the repository and connection broker.
6. Open the Services dialog box and verify that the application server was created correctly.  
   If the Documentum Java Method Server is started, it was created correctly.
7. Test the repository to verify that it is functioning correctly.
8. Shut down the repository on the first node.
9. Shut down the first node.
10. Start the second node.

11. Start the connection broker on the second node.

12. Upgrade the Content Server software on the second node. The Upgrading the Content Server Software, page 40 section provides the steps.

13. Start the configuration program and select Custom Configuration.

14. Select Upgrade and the repository to upgrade.

15. When the configuration program reaches the panel on which scripts are run, click Cancel.

   Do not run the scripts. The application server is created and the repository is upgraded.

16. To start the application server instance that is running the Java Method Server and ACS server, restart the Windows hosts after the upgrade is completed.

### Upgrading Content Server in a Distributed or Load-Balanced Configuration

Use the following procedure to upgrade the Content Server in a distributed or load-balanced configuration.

**To upgrade a distributed or load-balanced configuration:**

**Note:** On Windows, do not reboot the remote hosts by using Terminal Services. Reboot the remote hosts directly from those hosts.

1. On the primary host, upgrade Content Server, connection brokers, and repository.

2. On each remote host, upgrade Content Server and connection broker, but do not install a repository.

3. Run the cfsConfigurationProgram to Upgrade content-file server.

   **Note:** Make sure that you use the same key file name as the primary host.

4. To create an `acs config` object in the repository for each of the ACS servers installed with each remote Content Server (that is, if there are three remote Content Servers, create three `acs config` objects), run one of the following scripts on each remote Content Server host:

   - **Windows:** `%DM_HOME%\install\admin\dm_acs_install.ebs`
   - **Linux/AIX/Solaris:** `$DM_HOME/install/admin/dm_acs_install.ebs`

   The syntax is as follows

   ```
   dmabasic -f dm_acs_install.ebs -e Install -- repository_name user_name password
   acs_name server_config_name Java_method_server_port acs_protocol~CleanupCacheAcsObject
   CacheAcsDescriptionFile HostName
   ```

   **Table 8, page 46** describes the parameters. The `acs config` object is created in server config mode and uses the network locations, connection broker projection targets, and stores from the associated server config object. If you must change the mode to acs config mode, in which you manually set network locations, connection broker projection targets, and stores, use Documentum Administrator to change the mode and create the manual settings.

   **Note:** Do not use the Documentum API or DQL to modify the new acs config object.
### Table 8. Parameters required by dm_acs_install.ebs script

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description and values</th>
</tr>
</thead>
<tbody>
<tr>
<td>acs_name</td>
<td>Object name of the acs_config object you are creating. This name can be any arbitrary name, but the name must be unique among the object names of acs_config objects and the server_config objects of both the primary Content Server and any remote Content Server.</td>
</tr>
<tr>
<td>acs_protocol</td>
<td>Communication protocol used by the ACS server. Valid values are <strong>http</strong> and <strong>https</strong>.</td>
</tr>
<tr>
<td>CacheAcsDescriptionFile</td>
<td>File to store object dump of candidate dm_bocs_config objects. See CleanupCacheAcsObject.</td>
</tr>
<tr>
<td>CleanupCacheAcsObject</td>
<td>Set to F.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> T was only required when upgrading from a previous version from which upgrading is no longer supported.</td>
</tr>
<tr>
<td>HostName</td>
<td>Fully-qualified domain name of host. This value is used to set the acs_url attribute of the dm_acs_config used by the server named in dm_server_name.</td>
</tr>
<tr>
<td>Java_method_server_port</td>
<td>Port where the application server on the remote Content Server host listens, which was provided during remote Content Server installation. Do not change this port number after the initial configuration.</td>
</tr>
<tr>
<td>password</td>
<td>Password for the superuser account.</td>
</tr>
<tr>
<td>repository_name</td>
<td>Name of the repository served by the remote Content Server and its ACS server, where the acs_config object is being created.</td>
</tr>
<tr>
<td>server_config_name</td>
<td>Object name of the server_config object of the remote Content Server.</td>
</tr>
<tr>
<td>user_name</td>
<td>Username of a user with superuser privileges, for example, the installation owner.</td>
</tr>
</tbody>
</table>

5. If the remote Content Servers are installed in a different file-system path from the primary Content Server, create new site-specific location objects for locations that are new in the upgraded repository.

   a. Using Documentum Administrator, connect to the repository.

   b. Create site-specific `dm_dba` and `auth_plugin` location objects that contain the locations on each of the remote sites of `$DOCUMENTUM/dba` (Linux/AIX/Solaris) or `%DOCUMENTUM%\dba` (Windows) and the authentication plug-in.
c. In the server config object for the remote Content Server, set the `auth_plugin_location` and `dba_location` to the location objects you created.

6. Start the application server.

**Upgrading the BOCS Server**

A BOCS server is a caching server. It is a separate, optional product with its own installer. It is not installed with Content Server. The *EMC Documentum Content Server Distributed Configuration Guide* provides more information about BOCS.

The BOCS server is one component of a distributed configuration. When upgrading the Content Server in a distributed configuration, you may want to upgrade the BOCS server as well to use latest features available in BOCS/ACS.

**To upgrade the BOCS server:**

1. Log on as the owner of the existing BOCS installation.
2. Download and extract the compressed distribution file to a temporary location on the BOCS server host.
3. Stop the current BOCS server and its components.
4. Follow the installation steps provided in the “Installing BOCS” section in the *EMC Documentum Content Server Distributed Configuration Guide*.
5. Migrate `acs.properties` from the previous version’s location to the new location.

**Upgrading the DMS Server**

In a distributed configuration, the DMS server facilitates the precaching for BOCS server and asynchronous write operations for remote users. The *EMC Documentum Content Server Distributed Configuration Guide* provides more information about DMS. You can only upgrade DMS from 7.0 to 7.1 because in Documentum 7.0, the DMS installation was integrated with Content Server.

**To upgrade the DMS server:**

1. Ensure that you have Content Server 7.0 and DMS 7.0 installed in your environment.
2. Ensure that the connection broker and the global repository are configured for Content Server 7.0.
3. Upgrade Content Server 7.0 to 7.1. For the steps, see Upgrading the Content Server Software, page 40.
4. Upgrade the connection broker and global repository to 7.1.
5. Upgrade DMS to 7.1.
Upgrading Content Server

a. Use the Content Server configuration program to upgrade DMS:
   • Linux/AIX/Solaris: Run $DM_HOME/install/Server_Configuration_Program.sh and select Documentum Messaging Service (DMS).
   • Windows: Click Start > All Programs > Documentum > Documentum Server Manager and on the Utilities tab, click Server Configuration, and then select Documentum Messaging Service (DMS).

b. Complete the upgrade as instructed.

During the DMS upgrade, the old DMS installation is deleted from JBOSS 5.1.0 and DMS 7.1 is installed in JBOSS 7.1.1.

Post-Upgrade Tasks

After you upgrade Content Server to 7.x, perform the following tasks to improve the overall performance of Content Server.

Improving Performance after Upgrade

To avoid performance degradation after upgrading to Content Server 7.x, you must configure the values of the following two parameters:

• Number of method server threads: The method_server_threads parameter affects the number of worker threads. Depending on the number of dmbasic jobs, you can tune this value. For all operating systems, in the server.ini file, set method_server_threads = 3

• Java virtual machine (JVM) heap memory: The java.ini file specifies the options to the JVM, which is used by the dmbasic method server. This includes settings for the minimum and maximum heap memory for the JVM. Depending on your environment, you can tune this heap memory in the java.ini file:
   — For Window, set
     JAVA_OPTIONS=" -Xcheck:jni -XX:+RestoreMXCSROnJNICalls -Xms256m -Xmx512m"

   Xms specifies the startup heap value and Xmx specifies the maximum heap size for the Java heap. These values should be set or tuned cautiously to avoid allocating too much or too little heap memory. Allocating too much heap memory reduces the system’s memory for other processes. Allocating too little causes Java programs to crash with heap errors.

Extending the Oracle Tablespace Size

After you upgrade Content Server from 6.5 SP2 and later to 7.x, you must manually extend the Oracle tablespace size based on your requirements. This is because the default Oracle tablespace size, which is set to 2 GB, might be insufficient and may lead to tablespace-related issues in Documentum 7.x.
Enhancements after Upgrade

The 64-bit Content Server utilizes the underlying hardware and its components for better performance and scalability. All features are supported and work except those specifically mentioned below.

The Java Method Server runs 64-bit Java, which is bundled with the 64-bit Content Server installer. The dmbasic method server also uses the 64-bit Java Method Server.

Reinstalling D2 Client Libraries

D2 client JARS go missing on Content Server after Content Server is upgraded to 7.x. This occurs because the Content Server upgrade operation creates a new JBOSS folder, where the existing D2 client JARs are not preserved. Therefore, you must reinstall the D2 client JARS after upgrading the Content Server. The EMC Documentum D2 Installation Guide provides the steps for installing D2.

Rebuilding the database views

DQL statements are translated into SQL statements that query the database views. Content Server creates and maintains these views, which in some rare instances, may get corrupted. In such cases, after completing the upgrade process, you need to rebuild the database views. The views_valid attribute indicates the status of the views. It is part of dm_type object, so there is one views_valid attribute per Documentum object type. By setting the value of this attribute to false (0), you can force the Content Server to recreate the views.

1. Shut down the repository.
2. Connect to the database used by the repository through a SQL Editor. Connect as the repository owner.
3. Update the views_valid attribute for each corrupted object type using the following command:
   ```sql
   SQL> UPDATE dm_type_s SET views_valid = 0 WHERE name = 'dm_document';
   ```
4. If you are not sure which views to rebuild, you can rebuild the views for all the existing object types using the following command:
   ```sql
   SQL> UPDATE dm_type_s SET views_valid = 0;
   ```
5. Commit the changes in the database.
6. Restart the repository.

After the views are recreated, the views_valid attribute will be automatically set to true (1).
Chapter 4

Upgrade Scenarios

This chapter describes some of the supported scenarios for upgrading a previous version of Content Server to 7.2. Each scenario describes the upgrade path for the Content Server including the base and upgraded versions of the operating system, database, and Content Server, and the steps you need to perform for the upgrade.

Note: Although there can be multiple upgrade scenarios depending on the operating system/database combination, it is not possible to document all of those scenarios. This chapter only covers the upgrade of AIX/Oracle from 6.6 to 7.2

However, for a particular operating system/database combination, the upgrade steps do not vary much across Content Server versions. For example, if you are upgrading Content 6.6 on Linux/Oracle platform, use the upgrade steps covered in the “Upgrading Content Server 6.7 SP1 to 7.2 — Linux/Oracle” scenario.

This chapter contains the following topics:

- Upgrading Content Server 6.5 SP2 to 7.2 — Windows/SQL Server
- Upgrading Content Server 6.5 SP3 to 7.2 — AIX/DB2
- Upgrading Content Server 6.7 to 7.2 — Windows/Oracle
- Upgrading Content Server 6.7 SP1 to 7.2 — Linux/Oracle
- Upgrading Content Server 6.7 SP2 to 7.2 — Solaris/Oracle
- Upgrading Content Server 7.0 to 7.2 — Windows/SQL
- Upgrading Content Server 7.1 to 7.2 — Linux/Oracle

Upgrading Content Server 6.5 SP2 to 7.2 — Windows/SQL Server

This upgrade scenario provides the instructions for upgrading the 32-bit Content Server 6.5 SP2 installed on the Windows 2008 R2 (64-bit) operating system and using SQL Server 2008 SP1 as the database. The upgrade process involves a direct upgrade of the operating system to Windows Server 2008 R2 SP1 (64-bit) and the upgrading the database to SQL Server 2012. Finally, the 32-bit Content Server 6.5 SP2 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.
Table 9. Base and Upgraded OS/DB/CS Versions

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows 2008 R2 (64-bit)</td>
<td>Windows Server 2008 R2 SP1 (64-bit)</td>
</tr>
<tr>
<td>Database</td>
<td>SQL Server 2008 SP1</td>
<td>SQL Server 2012 SP1</td>
</tr>
<tr>
<td>Content Server</td>
<td>32-bit Content Server 6.5 SP2</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>

**Preupgrade Tasks**

1. Review the Chapter 3, Upgrading Content Server chapter.
2. If you are installing the xPlore indexing server, review the EMC Documentum xPlore Installation Guide.
3. Back up the repository.
4. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
5. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
6. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. For the steps, refer to the database documentation.
7. Ensure that you have sufficient disk space on the computer hosting the database.
8. Run the Consistency Checker tool. The syntax is:
   
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>  
9. Fix the inconsistencies reported by the Consistency Checker tool as errors.
10. Ensure that the dm_server_config object is unlocked.

**Upgrade Tasks**

1. Upgrade the Windows operating system to Windows Server 2008 R2 SP1 (64-bit).
2. Apply the SP2 patch to SQL Server 2008 SP1.
3. Upgrade the database to SQL Server 2012 SP1.
4. Update the DSN entry from 32-bit to 64-bit and verify that DSN is pointing to the 64-bit SQL libraries. The 64-bit ODBC and DSN Libraries, page 38 section provides more information about migrating the ODBC and DSN libraries from versions for 32-bit Content Server to versions for the 64-bit Content Server.
5. Disable all jobs in all repositories on the host.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. Right-click the job and select Properties.
   c. In the Properties window, set the State as Inactive.
6. Shut down the repositories and connection brokers.
a. Click Start > Programs > Documentum > Server Manager.

b. Select the correct Content Server and click Stop.

c. On the Connection Broker tab, select each connection broker, and then click Stop.

7. To shut down the application server on Windows, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.

- Click Start > Control Panel > Administrative Tools > Services, select the Java Method Server, and then click Stop.

8. During the upgrade from 32-bit to 64-bit, Content Server, you cannot upgrade the authentication plug-ins that you have installed. You need to replace the 32-bit authentication plug-ins with the 64-bit plug-ins. You can find the plug-ins in the %DM_HOME%\install\external_apps\authplugins folder.

9. Run the Content Server installation program.

a. Run serverSetup.exe (Windows) to launch the 64-bit Content Server Installer.

b. Click Yes when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.

c. Accept the license agreement and click Next.

d. Type the installation owner password and click Next.

e. Review the installation summary and click Install to begin installation.

f. Specify if you want to enter license keys for optional modules of Content Server and click Next.

- Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.

- No: You can always enter license keys for optional modules later using the Content Server configuration program.

g. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

10. Upgrade the connection broker.

a. In the Documentum Content Server Configuration Program wizard, select Connection broker and click Next.

b. Type the installation owner password and click Next.

c. Select Upgrade a connection broker.

d. Select the connection broker to upgrade from the list and click Next.

e. Complete the configuration, select Perform additional configuration, and click Next.

11. Upgrade the existing repository.

a. On the configuration program options page, select Repository and then click Next.

b. Select Upgrade an existing repository.

c. Select the repository to upgrade from the list, and click Next.
d. You can choose if you want to upgrade AEK key or continue with the existing AEK key. Take a backup of the AEK key. For creating new AEK key, refer to *EMC Documentum Content Server 7.2 Installation Guide*. By enabling the lockbox option, you can move the existing AEK key or the new AEK key into the lockbox. For more information on lockbox, refer to *EMC Documentum Content Server 7.2 Installation Guide*.

e. Type the **Connection Broker Port** and **Connection Broker Host** and click Next.

f. Select the **Connection Mode** for the repository and click Next. If you select the Secure or Native and Secure options, select **Use certificate** on the next page, provide the required details, and then click Next.

g. Type the **SMTP server name** and the **owner’s email address** and click Next.

h. Select the module you want to activate for the repository and click Next.

i. Specify whether you want to change the xDB superuser password and click Next.

j. Type the **Repository owner password** and the **Database administrator name** and **password**, and then click Next.

k. Select **Finish configuration** and click Next.

12. If you upgraded from an installation using FAST full-text indexing, and selected xplore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.

13. After you complete the Content Server configuration, create a nonunique index on the dm_sysobject.r_object_id and r_aspect_name properties by using the following **MAKE_INDEX** command:

   ```
   EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name', use_id_col=true
   ```

   The inclusion of the use_id_col argument set to true automatically causes Content Server to include the r_object_id column in the index.

14. If you are upgrading a repository in a distributed environment that uses a BOCS and asynchronous write jobs, create an index on the dmr_content_s.i_parked_state, r_object_id properties. Use the following **MAKE_INDEX** command to do so:

   ```
   EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state', use_id_col=true, id_in_front=false
   ```

**Post-Upgrade Tasks**

1. After the upgrade is complete, delete directory `%DOCUMENTUM%<OLD_JBOSS_HOME>`.

2. Enable all the disabled jobs.
   a. In Documentum Administrator, navigate to **Administrator > Job Management > Jobs**.
   b. For each of the previously disabled jobs, right-click the job and select **Properties**.
   c. In the **Properties** window, set the **State** option to **Active**.

3. On a Windows upgrade, the **Startup Type** is set to **Manual** for the Documentum Docbase Service repository name service. If you want the repository to automatically start after a server reboot, navigate to **Start > All Programs > Administrative Tools > Services**, and set the **Startup Type** to **Automatic**.
4. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete the upgrade.

5. Optionally, run dm_filestore_unique.class in %DM_HOME%\install\tools to create a filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

6. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure that you manually update the backed up customized attribute.

7. After the upgrade is complete, perform the following checks:
   a. Check whether the <Docbroker>.log file in the <Documentum_Home>\dba\log\folder contains any warning messages related to DM_DOCBROKER_W_SSL_HANDSHAKE FAILED.
   b. Check whether the <Docbase>.log file in the <Documentum_Home>\dba\log\folder contain any exceptions or errors.
   c. Check for error messages in the <JBOSS_Home>\server\DctmServer_MethodServer \log\server.log file.
   d. To check the Documentum version, in the command prompt, run the following command:
      
      ```
documentum -version
      ```
   e. To check the DFC version, ensure that the JRE bin path is set in the PATH variable, and then in the command prompt, run the following command:
      
      ```
java DfShowVersion
      ```
   f. You can also check the Documentum version mentioned in the repository and connection broker log files.

8. Review the Post-Upgrade Tasks, page 48 section for other post-upgrade tasks that you might need to perform.

### Upgrading Content Server 6.5 SP3 to 7.2 — AIX/DB2

This upgrade scenario provides the instructions for upgrading the 32-bit Content Server 6.5 SP3 installed on the IBM AIX 6.1 TL3 operating system and using DB2 9.7 Fix Pack 1 as the database. The upgrade process involves a direct upgrade of the operating system to IBM AIX 7.1 TL3 and upgrading the database to DB2 Enterprise 10.5 Fix Pack 3. Finally, the 32-bit Content Server 6.5 SP3 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.

**Table 10. Base and Upgraded OS/DB/CS Versions**

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>IBM AIX 6.1 TL3</td>
<td>IBM AIX 7.1 TL3</td>
</tr>
<tr>
<td>Database</td>
<td>DB2 9.7 Fix Pack 1</td>
<td>DB2 Enterprise 10.5 Fix Pack 3</td>
</tr>
<tr>
<td>Content Server</td>
<td>32-bit Content Server 6.5 SP3</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>
Preupgrade Tasks

1. Review the Chapter 3, Upgrading Content Server chapter.
2. Before upgrading Content Server on AIX with a DB2 database, create a temporary tablespace with an 8K page size. To create an 8K temporary tablespace, run the following command:
   \[ \text{db2 create temporary tablespace tempspace2 pagesize = 8192} \]
3. Backup all tables into the new tablespace.
4. Back up the repository.
5. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
6. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
7. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. For the steps, refer to the database documentation.
8. Ensure that you have sufficient disk space on the computer hosting the database.
9. Run the Consistency Checker tool. The syntax is:
   \[ \text{dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name>} \]
   \[ \text{<superuser> <password>} \]
10. Fix the inconsistencies reported by the Consistency Checker tool as errors.
11. Ensure that the dm_server_config object is unlocked.
12. Ensure that you perform the tasks mentioned in step 15 in the Upgrade Checklist, page 33.

Upgrade Tasks

1. Upgrade the AIX operating system to IBM AIX 7.1 TL3.
2. Upgrade the DB2 database to DB2 Enterprise 10.5 Fix Pack 3.
3. Disable all jobs in all repositories in the installation you are upgrading.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. Right-click the job and select Properties.
   c. In the Properties window, set the State as Inactive.
4. Shut down the repositories and connection brokers.
   a. For each repository, run the dm_shutdown_repository script, where repository is the name of the Content Server to be stopped.
   b. Stop each connection broker using the dm_stop_docbroker utility on the command line:
      \[ \% \text{dm_stop_docbroker [-Ppassword][-B[batch]} [-Nport_number][-Sservice_name]} \]
5. To shut down the application server on AIX, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
   - Run the script $DOCUMENTUM_SHARED/<JBOSS>/server/stopMethodServer.sh.
6. Run the Content Server installation program.
a. Run serverSetup.bin to launch the 64-bit Content Server Installer.
b. Click Yes when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.
c. Accept the license agreement and click Next.
d. Review the installation summary and click Install to begin installation.
e. Specify if you want to enter license keys for optional modules of Content Server and click Next.
   • Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.
   • No: You can always enter license keys for optional modules later using the Content Server configuration program.
f. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

7. Upgrade the connection broker.
   a. In the Documentum Content Server Configuration Program wizard, select Connection broker and click Next.
   b. Type the root password and the dmadmin group for running the $DOCUMENTUM/dba/dm_root_task script.
   c. Select Upgrade a connection broker.
   d. Select the connection broker to upgrade from the list and click Next.
   e. Complete the configuration, select Perform additional configuration, and click Next.

8. Upgrade the existing repository.
   a. On the configuration program options page, select Repository and then click Next.
   b. Select Upgrade an existing repository.
   c. Select the repository to upgrade from the list, and click Next.
   d. Type the Connection Broker Port and Connection Broker Host and click Next.
   e. Select the Connection Mode for the repository and click Next. If you select the Secure or Native and Secure options, select Use certificate on the next page, provide the required details, and then click Next.
   f. Select whether you want to create a global repository. If yes, specify the login name and password for the global repository.
   g. Select the modules you want to activate for the repository and click Next.
   h. Specify whether you want to change the xDB superuser password and click Next.
   i. Type the Repository owner password and the Database administrator name and password, and then click Next.
   j. Select Finish configuration and click Next.
9. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.

10. After you complete the Content Server configuration, create a nonunique index on the dm_sysobject.r_object_id and r_aspect_name properties by using the following MAKE_INDEX command:

   EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name',
   use_id_col=true

   The inclusion of the use_id_col argument set to true automatically causes Content Server to include the r_object_id column in the index.

11. If you are upgrading a repository in a distributed environment that uses a BOCS and asynchronous write jobs, create an index on the dmr_content_s (i_parked_state, r_object_id) properties. Use the following MAKE_INDEX command to do so:

   EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state',
   use_id_col=true, id_in_front=false

**Post-Upgrade Tasks**

1. After the upgrade is complete, delete the directory $DOCUMENTUM_SHARED/<OLD_JBOSS_HOME>.

2. Enable all the disabled jobs.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. For each of the previously disabled jobs, right-click the job and select Properties.
   c. In the Properties window, set the State option to Active.

3. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete the upgrade.

4. Optionally, run dm_filestore_unique.class in $DM_HOME/install/tools to create a filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

5. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure that you manually update the backed up customized attribute.

6. After the upgrade is complete, perform the following checks:
   a. Check whether the <Docbroker>.log file in the <Documentum_Home>/dba/log/ folder contains any warning messages related to DM_DOCBROKER_W_SSL_HANDSHAKE_FAILED.
   b. Check whether the <Docbase>.log file in the <Documentum_Home>/dba/log/ folder contain any exceptions or errors.
   d. To check the Documentum version, in the command prompt, run the following command:
      ```
      documentum -version
      ```
   e. To check the DFC version, ensure that the JRE bin path is set in the PATH variable, and then in the command prompt, run the following command:
      ```
      java DfShowVersion
      ```
You can also check the Documentum version mentioned in the repository and connection broker log files.

7. Review the Post-Upgrade Tasks, page 48 section for other post-upgrade tasks that you might need to perform.

## Upgrading Content Server 6.5 SP3 to 7.2 — Windows/SQL Server

This upgrade scenario provides the instructions for upgrading the 32-bit Content Server 6.5 SP3 installed on the Windows 2008 R2 (64-bit) operating system and using SQL Server 2008 SP1 as the database. The upgrade process involves a direct upgrade of the operating system to Windows Server 2008 R2 SP1 (64-bit) and the upgrading the database to SQL Server 2012 SP1. Finally, the 32-bit Content Server 6.6 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows 2008 R2 (64-bit)</td>
<td>Windows Server 2008 R2 SP1 (64-bit)</td>
</tr>
<tr>
<td>Database</td>
<td>SQL Server 2008 SP1</td>
<td>SQL Server 2012 SP1</td>
</tr>
<tr>
<td>Content Server</td>
<td>32-bit Content Server 6.5 SP3</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>

### Preupgrade Tasks

1. Review the Chapter 3, Upgrading Content Server chapter.
2. If you are installing the xPlore indexing server, review the EMC Documentum xPlore Installation Guide.
3. Back up the repository.
4. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
5. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
6. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. For the steps, refer to the database documentation.
7. Ensure that you have sufficient disk space on the computer hosting the database.
8. Run the Consistency Checker tool. The syntax is:
   ```
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>
   ```
9. Fix the inconsistencies reported by the Consistency Checker tool as errors.
10. Ensure that the `dm_server_config` object is unlocked.

**Upgrade Tasks**

1. Upgrade the Windows operating system to Windows Server 2008 R2 SP1 (64-bit).
2. Apply the SP2 patch to SQL Server 2008 SP1.
3. Upgrade the database to SQL Server 2012 SP1.
4. Update the DSN entry from 32-bit to 64-bit and verify that DSN is pointing to the 64-bit SQL libraries. The *64-bit ODBC and DSN Libraries*, page 38 section provides more information about migrating the ODBC and DSN libraries from versions for 32-bit Content Server to versions for the 64-bit Content Server.
5. Disable all jobs in all repositories on the host.
   a. In Documentum Administrator, navigate to **Administrator > Job Management > Jobs**.
   b. Right-click the job and select **Properties**.
   c. In the Properties window, set the **State** as **Inactive**.
6. Shut down the repositories and connection brokers.
   a. Click **Start > Programs > Documentum > Server Manager**.
   b. Select the correct Content Server and click **Stop**.
   c. On the **Connection Broker** tab, select each connection broker, and then click **Stop**.
7. To shut down the application server on Windows, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
   • Click **Start > Control Panel > Administrative Tools > Services**, select the **Java Method Server**, and then click **Stop**.
8. During the upgrade from 32-bit to 64-bit, Content Server, you cannot upgrade the authentication plug-ins that you have installed. You need to replace the 32-bit authentication plug-ins with the 64-bit plug-ins. You can find the plug-ins in the `%DM_HOME%\install\external_apps\authplugins` folder.
9. Run the Content Server installation program.
   a. Run `serverSetup.exe` (Windows) to launch the 64-bit Content Server Installer.
   b. Click **Yes** when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.
   c. Accept the license agreement and click **Next**.
   d. Type the **installation owner password** and click **Next**.
   e. Review the installation summary and click **Install** to begin installation.
f. Specify if you want to enter license keys for optional modules of Content Server and click Next.
   • Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.
   • No: You can always enter license keys for optional modules later using the Content Server configuration program.

g. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

10. Upgrade the connection broker.
   a. In the Documentum Content Server Configuration Program wizard, select Connection broker and click Next.
   b. Type the installation owner password and click Next.
   c. Select Upgrade a connection broker.
   d. Select the connection broker to upgrade from the list and click Next.
   e. Complete the configuration, select Perform additional configuration, and click Next.

11. Upgrade the existing repository.
   a. On the configuration program options page, select Repository and then click Next.
   b. Select Upgrade an existing repository.
   c. Select the repository to upgrade from the list, and click Next.
   d. You can choose if you want to upgrade AEK key or continue with the existing AEK key. Take a backup of the AEK key. For creating new AEK key, refer to EMC Documentum Content Server 7.2 Installation Guide.
       By enabling the lockbox option, you can move the existing AEK key or the new AEK key into the lockbox. For more information on lockbox, refer to EMC Documentum Content Server 7.2 Installation Guide.
   e. Type the Connection Broker Port and Connection Broker Host and click Next.
   f. Select the Connection Mode for the repository and click Next. If you select the Secure or Native and Secure options, select Use certificate on the next page, provide the required details, and then click Next.
   g. Type the SMTP server name and the owner’s email address and click Next.
   h. Select the module you want to activate for the repository and click Next.
   i. Specify whether you want to change the xDB superuser password and click Next.
   j. Type the Repository owner password and the Database administrator name and password, and then click Next.
   k. Select Finish configuration and click Next.

12. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.
13. After you complete the Content Server configuration, create a nonunique index on the
   `dm_sysobject.r_object_id` and `r_aspect_name` properties by using the following
   `MAKE_INDEX` command:
   
   ```
   EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name',
   use_id_col=true
   
   The inclusion of the use_id_col argument set to true automatically causes Content Server to
   include the r_object_id column in the index.
   ```

14. If you are upgrading a repository in a distributed environment that uses a BOCS and
    asynchronous write jobs, create an index on the `dmr_content_s(i_parked_state,
    r_object_id)` properties. Use the following `MAKE_INDEX` command to do so:
    
    ```
    EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state',
    use_id_col=true, id_in_front=false
    ```

**Post-Upgrade Tasks**

1. After the upgrade is complete, delete directory `%DOCUMENTUM%\OLD_JBOSS_HOME`.

2. Enable all the disabled jobs.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. For each of the previously disabled jobs, right-click the job and select Properties.
   c. In the Properties window, set the State option to Active.

3. On a Windows upgrade, the Startup Type is set to Manual for the Documentum Docbase Service
    repository name service. If you want the repository to automatically start after a server reboot,
    navigate to Start > All Programs > Administrative Tools > Services, and set the Startup Type
    to Automatic.

4. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete
    the upgrade.

5. Optionally, run `dm_filestore_unique.class` in `%DM_HOME%\install\tools` to create a
    filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

6. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure
    that you manually update the backed up customized attribute.

7. After the upgrade is complete, perform the following checks:
   a. Check whether the `<Docbroker>.log` file in the `<Documentum_Home>\dba\log\folder`
      contains any warning messages related to DM_DOCBROKER_W_SSL_HANDSHAKE
      FAILED.
   b. Check whether the `<Docbase>.log` file in the `<Documentum_Home>\dba\log\folder`
      contain any exceptions or errors.
   c. Check for error messages in the `<JBoss_Home>\server\DctmServer_MethodServer
      \log\server.log` file.
   d. To check the Documentum version, in the command prompt, run the following command:
      ```
      documentum -version
      ```
   e. To check the DFC version, ensure that the JRE bin path is set in the PATH variable, and then
      in the command prompt, run the following command:
      ```
      java DfShowVersion
      ```
You can also check the Documentum version mentioned in the repository and connection broker log files.

Review the Post-Upgrade Tasks, page 48 section for other post-upgrade tasks that you might need to perform.

**Upgrading Content Server 6.7 to 7.2 — Windows/Oracle**

This upgrade scenario provides the instructions for upgrading the 64-bit Content Server 6.7 installed on the Windows 2008 R2 (64-bit) operating system and using Oracle 11g (11.2.0.1) as the database. The upgrade process involves a direct upgrade of the operating system to Windows Server 2008 R2 SP1 (64-bit) and the upgrading the database to Oracle 12c (12.1.0.2). Finally, the 64-bit Content Server 6.7 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.

<table>
<thead>
<tr>
<th>Table 12. Base and Upgraded OS/DB/CS Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base Version</strong></td>
</tr>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Content Server</td>
</tr>
</tbody>
</table>

**Preupgrade Tasks**

Follow the steps provided in the Preupgrade Tasks, page 59 section.

**Upgrade Tasks**

1. Upgrade the Windows operating system to Windows Server 2012 (64-bit).
2. Upgrade the Oracle database to Oracle 12c (12.1.0.2).
3. After you upgrade the database, create an ORACLE_HOME environment variable in Windows that points to the location of the 64-bit tnsnames.ora file. The entries from the 32-bit tnsnames.ora file have to be copied into the 64-bit tnsnames.ora file.
4. Follow steps 5 - 14 in the Upgrade Tasks, page 60 section.

**Post-Upgrade Tasks**

Follow the steps provided in the Post-Upgrade Tasks, page 62 section.
Upgrading Content Server 6.7 SP1 to 7.2 — Linux/Oracle

This upgrade scenario provides the instructions for upgrading the 32-bit Content Server 6.7 SP1 installed on the Red Hat Enterprise Linux 6.1 (64-bit) operating system and using Oracle 11g (11.2.0.3) as the database. The upgrade process involves a direct upgrade of the operating system to Red Hat Enterprise Linux 6.5 (64-bit) and upgrading the database to Oracle 12c (12.1.0.2). Finally, the 32-bit Content Server 6.7 SP1 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.

### Table 13. Base and Upgraded OS/DB/CS Versions

<table>
<thead>
<tr>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux 6.1 (64-bit)</td>
<td>Red Hat Enterprise Linux 6.5 (64-bit)</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 6.1 (64-bit)</td>
<td>Red Hat Enterprise Linux 7.0 (64-bit) [2]</td>
</tr>
<tr>
<td>SUSE Linux Enterprise 11.1 (64-bit)</td>
<td>SUSE Linux Enterprise 12 (64-bit) [1]</td>
</tr>
<tr>
<td>Oracle 11g (11.2.0.3)</td>
<td>Oracle 12c (12.1.0.2)</td>
</tr>
</tbody>
</table>

[1] You cannot perform the upgrade from SUSE Linux Enterprise 11.1 to SUSE Linux Enterprise 12 directly. This upgrade process consists of two steps:
- Perform the upgrade from SUSE Linux Enterprise 11.1 to SUSE Linux Enterprise 11.3.
- Perform the upgrade from SUSE Linux Enterprise 11.3 to SUSE Linux Enterprise 12.

[2] You cannot perform the upgrade from Red Hat Enterprise Linux 6.1 to Red Hat Enterprise Linux 7.0 directly. This upgrade process consists of two steps:
- Perform the upgrade from Red Hat Enterprise Linux 6.1 to Red Hat Enterprise Linux 6.5.
- Perform the upgrade from Red Hat Enterprise Linux 6.5 to Red Hat Enterprise Linux 7.0.

### Preupgrade Tasks

1. Review the Chapter 3, Upgrading Content Server chapter.
2. Back up the repository.
3. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
4. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
5. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. For the steps, refer to the database documentation.

6. Ensure that you have sufficient disk space on the computer hosting the database.

7. Run the Consistency Checker tool. The syntax is:
   
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>

8. Fix the inconsistencies reported by the Consistency Checker tool as errors.

9. Ensure that the dm_server_config object is unlocked.

10. Ensure that you perform the tasks mentioned in step 15 in the Upgrade Checklist, page 33.

Upgrade Tasks

1. Upgrade the Linux operating system to Red Hat Enterprise Linux 6.5 (64-bit)/ SUSE Enterprise Linux 12 (64-bit)

2. Upgrade the Oracle database to Oracle 12c (12.1.0.2).

3. Disable all jobs in all repositories in the installation you are upgrading.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. Right-click the job and select Properties.
   c. In the Properties window, set the State as Inactive.

4. Shut down the repositories and connection brokers.
   a. For each repository, run the dm_shutdown_repository script, where repository is the name of the Content Server to be stopped.
   b. Stop each connection broker using the dm_stop_docbroker utility on the command line:

   % dm_stop_docbroker [-Ppassword] [-B[batch]] [-Nport_number] [-Sservice_name]

5. To shut down the application server on Linux, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
   - Run the script $DOCUMENTUM_SHARED/<JBOSS>/server/stopMethodServer.sh.

6. Run the Content Server installation program.
   a. Run serverSetup.bin to launch the 64-bit Content Server Installer.
   b. Click Yes when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.
   c. Accept the license agreement and click Next.
   d. In the Root User Password box, type the password, and click Next.
   e. Review the installation summary and click Install to begin installation.
f. Specify if you want to enter license keys for optional modules of Content Server and click Next.
   - Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.
   - No: You can always enter license keys for optional modules later using the Content Server configuration program.

g. To launch the Content Server configuration program and configure the repository, select **Configure now** and click **Done**.

7. Upgrade the connection broker.
   a. In the Documentum Content Server Configuration Program wizard, select **Connection broker** and click **Next**.
   b. Select **Upgrade a connection broker**.
   c. Select the connection broker to upgrade from the list and click **Next**.
   d. Complete the configuration, select **Perform additional configuration**, and click **Next**.

8. Upgrade the existing repository.
   a. On the configuration program options page, select **Repository** and then click **Next**.
   b. Select **Upgrade an existing repository**.
   c. Select the repository to upgrade from the list, and click **Next**.
   d. You can choose if you want to upgrade AEK key or continue with the existing AEK key. Take a backup of the AEK key. For creating new AEK key, refer to *EMC Documentum Content Server 7.2 Installation Guide*.

      By enabling the lockbox option, you can move the existing AEK key or the new AEK key into the lockbox. For more information on lockbox, refer to *EMC Documentum Content Server 7.2 Installation Guide*.
   e. Type the **Connection Broker Port** and **Connection Broker Host** and click **Next**.
   f. Select the **Connection Mode** for the repository and click **Next**. If you select the **Secure** or **Native and Secure** options, select **Use certificate** on the next page, provide the required details, and then click **Next**.
   g. Select the modules you want to activate for the repository and click **Next**.
   h. Specify whether you want to change the xDB superuser password and click **Next**.
   i. Type the **Repository owner password** and the **Database administrator name** and **password**, and then click **Next**.
   j. Select **Finish configuration** and click **Next**.

9. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.

10. After you complete the Content Server configuration, create a nonunique index on the `dm_sysobject.r_object_id` and `r_aspect_name` properties by using the following `MAKE_INDEX` command:
EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name', use_id_col=true

The inclusion of the use_id_col argument set to true automatically causes Content Server to include the r_object_id column in the index.

11. If you are upgrading a repository in a distributed environment that uses a BOCS and asynchronous write jobs, create an index on the dmr_content_s (i_parked_state, r_object_id) properties. Use the following MAKE_INDEX command to do so:

EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state', use_id_col=true, id_in_front=false

Post-Upgrade Tasks

1. After the upgrade is complete, delete the directory $DOCUMENTUM_SHARED/<OLD_JBOSS_HOME>.

2. Enable all the disabled jobs.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. For each of the previously disabled jobs, right-click the job and select Properties.
   c. In the Properties window, set the State option to Active.

3. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete the upgrade.

4. Optionally, run dm_filestore_unique.class in $DM_HOME/install/tools to create a filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

5. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure that you manually update the backed up customized attribute.

6. After the upgrade is complete, perform the following checks:
   a. Check whether the <Docbroker>.log file in the <Documentum Home>/dba/log/ folder contains any warning messages related to DM_DOCBROKER_W_SSL_HANDSHAKE_FAILED.
   b. Check whether the <Docbase>.log file in the <Documentum Home>/dba/log/ folder contain any exceptions or errors.
   d. To check the Documentum version, in the command prompt, run the following command:
      documentum -version
   e. To check the DFC version, ensure that the JRE bin path is set in the PATH variable, and then in the command prompt, run the following command:
      java DfShowVersion
   f. You can also check the Documentum version mentioned in the repository and connection broker log files.

7. Review the Post-Upgrade Tasks, page 48 section for other post-upgrade tasks that you might need to perform.
Upgrading Content Server 6.7 SP2 to 7.2 — Solaris/Oracle

This upgrade scenario provides the instructions for upgrading the 32-bit Content Server 6.7 SP2 installed on the Oracle Solaris 11 (SPARC) operating system and using Oracle 11g (11.2.0.3) as the database. The upgrade process involves a direct upgrade of the 32-bit Content Server 6.7 SP2 to 64-bit Content Server 7.2. You have to upgrade the operating system to Oracle Solaris 11.2 (SPARC) & database to Oracle 12c (12.1.0.2).

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.

### Table 14. Base and Upgraded OS/DB/CS Versions

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Oracle Solaris 11 (SPARC)</td>
<td>Oracle Solaris 11.2 (SPARC)</td>
</tr>
<tr>
<td>Database</td>
<td>Oracle 11g (11.2.0.3.0)</td>
<td>Oracle 12c (12.1.0.2)</td>
</tr>
<tr>
<td>Content Server</td>
<td>32-bit Content Server 6.7 SP2</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>

### Preupgrade Tasks

1. Review the Chapter 3, Upgrading Content Server chapter.
2. If you are installing the xPlore indexing server, review the EMC Documentum xPlore Installation Guide.
3. Back up the repository.
4. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
5. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
6. Run the Consistency Checker tool. The syntax is:
   ```
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>
   ```
7. Fix the inconsistencies reported by the Consistency Checker tool as errors.
8. Ensure that you perform the tasks mentioned in step 15 in the Upgrade Checklist, page 33.

### Upgrade Tasks

1. Disable all jobs in all repositories in the installation you are upgrading.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. Right-click the job and select Properties.
   c. In the Properties window, set the State as Inactive.
2. Shut down the repositories and connection brokers.
a. For each repository, run the `dm_shutdown_repository` script, where repository is the name of the Content Server to be stopped.

b. Stop each connection broker using the `dm_stop_docbroker` utility on the command line:

   % dm_stop_docbroker [-Ppassword][-B[batch]] [-Nport_number][-Sservice_name]

3. To shut down the application server on Solaris, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
   • Run the script `$DOCUMENTUM_SHARED/<JBOSS>/server/stopMethodServer.sh`.

4. During the upgrade from 32-bit to 64-bit, Content Server, you cannot upgrade the authentication plug-ins that you have installed. You need to replace the 32-bit authentication plug-ins with the 64-bit plug-ins. You will find the plug-ins in the `%DM_HOME%/install/external_apps/authplugins` folder.

5. Run the Content Server installation program.
   a. Run `serverSetup.bin` to launch the 64-bit Content Server Installer.
   b. Click Yes when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.
   c. Accept the license agreement and click Next.
   d. Type the installation owner password and click Next.
   e. Review the installation summary and click Install to begin installation.
   f. Specify if you want to enter license keys for optional modules of Content Server and click Next.
      • Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.
      • No: You can always enter license keys for optional modules later using the Content Server configuration program.
   g. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

6. Upgrade the connection broker.
   a. In the Documentum Content Server Configuration Program wizard, select Connection broker and click Next.
   b. Select Upgrade a connection broker.
   c. Select the connection broker to upgrade from the list and click Next.
   d. Complete the configuration, select Perform additional configuration, and click Next.

7. Upgrade the existing repository.
   a. On the configuration program options page, select Repository and then click Next.
   b. Select Upgrade an existing repository.
   c. Select the repository to upgrade from the list, and click Next.
d. You can choose if you want to upgrade AEK key or continue with the existing AEK key. Take a backup of the AEK key. For creating new AEK key, refer to EMC Documentum Content Server 7.2 Installation Guide.

By enabling the lockbox option, you can move the existing AEK key or the new AEK key into the lockbox. For more information on lockbox, refer to EMC Documentum Content Server 7.2 Installation Guide.

e. Type the Connection Broker Port and Connection Broker Host and click Next.

f. Select the Connection Mode for the repository and click Next. If you select the Secure or Native and Secure options, select Use certificate on the next page, provide the required details, and then click Next.

g. Specify the XML Store port, the XML Store directory location, the page size, and then click Next.

h. Select the modules you want to activate for the repository and click Next.

i. Specify whether you want to change the xDB superuser password and click Next.

j. Type the Repository owner password and the Database administrator name and password, and then click Next.

k. Select Finish configuration and click Next.

8. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.

9. After you complete the Content Server configuration, create a nonunique index on the dm_sysobject.r_object_id and r_aspect_name properties by using the following MAKE_INDEX command:

```
EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name', use_id_col=true
```

The inclusion of the use_id_col argument set to true automatically causes Content Server to include the r_object_id column in the index.

10. If you are upgrading a repository in a distributed environment that uses a BOCs and asynchronous write jobs, create an index on the dmr_content_s(i_parked_state, r_object_id) properties. Use the following MAKE_INDEX command to do so:

```
EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state', use_id_col=true,id_in_front=false
```

### Post-Upgrade Tasks

1. After the upgrade is complete, delete the directory $DOCUMENTUM_SHARED/<OLD_JBOSS_HOME>.

2. Enable all the disabled jobs.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. For each of the previously disabled jobs, right-click the job and select Properties.
   c. In the Properties window, set the State option to Active.

3. Set the JAVA_HOME and PATH environment variables to the JDK 1.7 directory after you complete the upgrade.
4. Optionally, run `dm_filestore_unique.class` in `$DM_HOME/install/tools` to create a filestore lock file after upgrade. Processing result (success or failure) can be found in the log file.

5. After upgrade, ensure that you add the backed up list of users to the Admin group. Also, ensure that you manually update the backed up customized attribute.

6. After the upgrade is complete, perform the following checks:
   a. Check whether the `<Docbroker>.log` file in the `<Documentum_Home>/dba/log/` folder contains any warning messages related to `DM_DOCBROKER_W_SSL_HANDSHAKE_FAILED`.
   b. Check whether the `<Docbase>.log` file in the `<Documentum_Home>/dba/log/` folder contain any exceptions or errors.
   d. To check the Documentum version, in the command prompt, run the following command:
      ```
documentum -version
```
   e. To check the DFC version, ensure that the JRE bin path is set in the `PATH` variable, and then in the command prompt, run the following command:
      ```
java DfShowVersion
```
   f. You can also check the Documentum version mentioned in the repository and connection broker log files.

7. Run the Federation job.
   a. Log in to Documentum Administrator as the repository owner of the governing repository.
   b. Set the connection broker to all machines on which the repositories are installed.
   c. Add more repositories by providing the machine IP addresses and connection broker ports.
   d. To create a Federation, navigate to Configuration > Federations.
   e. Click File > New > Federation.
   f. Select the governing repository and click Next.
   g. Provide the repository owner name and password and then click Next.
   h. Provide a name for the Federation and then click Next. Click Next.
   i. To add member repositories, click the Members tab to access the Federation Configuration Properties - Members page.
   j. To add a member repository, click Add to access the Choose Member Repositories page.
   k. Locate the repository that you want to add, select the repository name, then click Add and OK.
   l. Type the name and password of a user who has superuser privileges in the new member repository and click OK.
   m. Click Finish.
   n. Set `$DM_HOME/bin` in your classpath.
   o. Restart the machine.
p. Create a user in the global repository and run the `dm_FederationUpdate` script.
q. Confirm the successful creation of the Federation by checking if the newly created user has been propagated to the member repositories.

8. Run the Replication job.
   a. Ensure that there are two repositories that are running to perform replication.
   b. In Documentum Administrator, navigate to File > New > Replication Job.
   c. In the New Replication Job window, type a name for the job.
   d. Set the Trace Level to 10 and then click Next.
   e. On the Schedule tab, click Next.
   f. On the From Source tab, provide the administrator credentials, set the source repository, specify the source repository path, and then click Next.
   g. On the To Target tab, provide the administrator credentials, set the target repository, specify the target repository path, and then click Next.
   h. On the Replication Options tab, click Next.
   i. On the SysObject Info tab, provide the title and subject and then click Finish.
   j. Under Jobs, right-click the newly created job and click Run.
   k. After the replication job completes, verify that all data has been replicated from source to the target repository.

9. Run the LDAP sync job.
   a. In Documentum Administrator, navigate to Administration > Job Management > Jobs.
   b. Search for `dm_LDAPSync`, select it, and then click Run.

10. Review the Post-Upgrade Tasks, page 48 section for other post-upgrade tasks that you might need to perform.

**Upgrading Content Server 7.0 to 7.2 — Windows/SQL**

This upgrade scenario provides the instructions for upgrading the 64-bit Content Server 7.0 installed on the Windows Server 2008 R2 SP1 (64-bit) operating system and using SQL Server 2012 as the database. The upgrade process involves a direct upgrade of the operating system to Windows Server 2012 R2 (64-bit) and the upgrading the database to SQL Server 2012 SP2. Finally, the 64-bit Content Server 7.0 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.
Table 15. Base and Upgraded OS/DB/CS Versions

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows Server 2008 R2 SP1 (64-bit)</td>
<td>Windows Server 2012 R2 (64-bit)</td>
</tr>
<tr>
<td>Database</td>
<td>SQL Server 2012</td>
<td>SQL Server 2012 SP2</td>
</tr>
<tr>
<td>Content Server</td>
<td>64-bit Content Server 7.0</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>

Preupgrade Tasks

1. Review the [Chapter 3, Upgrading Content Server](#) chapter.
2. Back up the repository.
3. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
4. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
5. Temporarily increase the amount of rollback space available in the RDBMS. The number of rollback segments should be commensurate with the size of the repository and should be in segments of equal size. For the steps, refer to the database documentation.
6. Ensure that you have sufficient disk space on the computer hosting the database.
7. Run the Consistency Checker tool. The syntax is:

   ```
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name>
   <superuser> <password>
   ```

8. Fix the inconsistencies reported by the Consistency Checker tool as errors.
9. Ensure that the dm_server_config object is unlocked.

Upgrade Tasks

1. Upgrade the Windows operating system to Windows Server 2012 R2 (64-bit).
2. Upgrade the database to SQL Server 2012 SP2.
3. Disable all jobs in all repositories on the host.
   a. In Documentum Administrator, navigate to Administrator > Job Management > Jobs.
   b. Right-click the job and select Properties.
   c. In the Properties window, set the State as Inactive.
4. Shut down the repositories and connection brokers.
   a. Click Start > Programs > Documentum > Server Manager.
   b. Select the correct Content Server and click Stop.
   c. On the Connection Broker tab, select each connection broker, and then click Stop.
5. To shut down the application server on Windows, stop the service called Documentum Java Method Server. Ensure that the application server does not start automatically after a host restart.
   • Click Start > Control Panel > Administrative Tools > Services, select the Java Method Server, and then click Stop.

6. Run the Content Server installation program.
   a. Run serverSetup.exe (Windows) to launch the 64-bit Content Server Installer.
   b. Click Yes when the installer displays a message stating that you are trying to upgrade the older version and asks if you want to proceed.
   c. Accept the license agreement and click Next.
   d. Type the installation owner password and click Next.
   e. Review the installation summary and click Install to begin installation.
   f. Specify if you want to enter license keys for optional modules of Content Server and click Next.
      • Yes: On the next page, select the optional modules you want to install and enter corresponding license keys. You can enable extended services such as Trusted Content Services, Content Services for EMC Centera, Content Storage Services, and so on.
      • No: You can always enter license keys for optional modules later using the Content Server configuration program.
   g. To launch the Content Server configuration program and configure the repository, select Configure now and click Done.

7. Upgrade the connection broker.
   a. In the Documentum Content Server Configuration Program wizard, select Connection broker and click Next.
   b. Type the installation owner password and click Next.
   c. Select Upgrade a connection broker.
   d. Select the connection broker to upgrade from the list and click Next.
   e. Complete the configuration, select Perform additional configuration, and click Next.

8. Upgrade the existing repository.
   a. On the configuration program options page, select Repository and then click Next.
   b. Select Upgrade an existing repository.
   c. Select the repository to upgrade from the list, and click Next.
   d. You can choose if you want to upgrade AEK key or continue with the existing AEK key. Take a backup of the AEK key. For creating new AEK key, refer to EMC Documentum Content Server 7.2 Installation Guide.
      By enabling the lockbox option, you can move the existing AEK key or the new AEK key into the lockbox. For more information on lockbox, refer to EMC Documentum Content Server 7.2 Installation Guide.
   e. Type the Connection Broker Port and Connection Broker Host and click Next.
f. Select the **Connection Mode** for the repository and click **Next**. If you select the **Secure** or **Native and Secure** options, select **Use certificate** on the next page, provide the required details, and then click **Next**.

g. Type the **SMTP server name** and the **owner’s email address** and click **Next**.

h. Select the module you want to activate for the repository and click **Next**.

i. Specify whether you want to change the xDB superuser password and click **Next**.

j. Type the **Repository owner password** and the **Database administrator name** and **password**, and then click **Next**.

k. Select **Finish configuration** and click **Next**.

9. If you upgraded from an installation using FAST full-text indexing, and selected xPlore full-text indexing, you must restart Content Server again after you complete the Content Server configuration.

10. After you complete the Content Server configuration, create a nonunique index on the **dm_sysobject.r_object_id** and **r_aspect_name** properties by using the following **MAKE_INDEX** command:

    ```
    EXECUTE make_index WITH type_name='dm_sysobject', attribute='r_aspect_name', use_id_col=true
    ```

    The inclusion of the use_id_col argument set to true automatically causes Content Server to include the r_object_id column in the index.

11. If you are upgrading a repository in a distributed environment that uses a BOCS and asynchronous write jobs, create an index on the **dmr_content_s(i_parked_state, r_object_id)** properties. Use the following **MAKE_INDEX** command to do so:

    ```
    EXECUTE make_index WITH type_name='dmr_content', attribute='i_parked_state', use_id_col=true,id_in_front=false
    ```

**Post-Upgrade Tasks**

Follow the steps provided in the **Post-Upgrade Tasks, page 62** section.

---

**Upgrading Content Server 7.1 to 7.2 — Linux/Oracle**

This upgrade scenario provides the instructions for upgrading the 64-bit Content Server 7.1 installed on the Red Hat Enterprise Linux 6.4 (64-bit) operating system and using Oracle 11g (11.2.0.3.0) as the database. The upgrade process involves a direct upgrade of the operating system to Red Hat Enterprise Linux 6.5 (64-bit) and upgrading the database to Oracle 12c (12.1.0.2). Finally, the 64-bit Content Server 7.1 is upgraded to 64-bit Content Server 7.2.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this upgrade scenario.
<table>
<thead>
<tr>
<th>Table 16. Base and Upgraded OS/DB/CS Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 6.4 (64-bit)</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 6.4 (64-bit)</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Content Server</td>
</tr>
</tbody>
</table>

[1] You cannot perform the upgrade from Red Hat Enterprise Linux 6.4 to Red Hat Enterprise Linux 7.0 directly. This upgrade process consists of two steps:

- Perform the upgrade from Red Hat Enterprise Linux 6.4 to Red Hat Enterprise Linux 6.5.
- Perform the upgrade from Red Hat Enterprise Linux 6.5 to Red Hat Enterprise Linux 7.0.

**Preupgrade Tasks**

1. Review the Chapter 3, Upgrading Content Server chapter.
2. Back up the repository.
3. Optionally, you can take a backup of all users that are part of the Admin group. Also, ensure that you take the backup of the customized attribute like group_address.
4. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
5. Run the Consistency Checker tool. The syntax is:
   ```
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>
   ```
6. Fix the inconsistencies reported by the Consistency Checker tool as errors.
7. Ensure that the dm_server_config object is unlocked.
8. Ensure that you perform the tasks mentioned in step 15 in the Upgrade Checklist, page 33.

**Upgrade Tasks**

Follow the steps 3 through 11 in the Upgrade Tasks, page 65 section.

**Post-Upgrade Tasks**

Follow the steps provided in the Post-Upgrade Tasks, page 67 section.
Chapter 5

Migrating Content Server

Installing a new instance of Content Server 7.x, and migrating data from a previous version on a separate host, requires a procedure different from an upgrade.

This section addresses any variation from the basic scenario to known issues surrounding the configuration of your Documentum 7.x Content Server.

This chapter contains the following topics:

• Understanding the Migration Process, page 77
• Migration Checklist, page 79
• Planning a Migration, page 80
• Migration Methods, page 81
• Migrating Data Using SQL Server, page 85
• Migrating Data from Earlier Versions of Content Server, page 90
• Migrating XML Content to the XML Store, page 90
• Using DQL to Migrate Content to an XML Store, page 91
• Migrating Custom Content Server Methods, page 91
• Migrating DocApps and BOF2 Modules, page 91

Understanding the Migration Process

Content Server migration involves three phases:

• Migrating the 32-bit Content Server to the platform running a 64-bit operating system.
• Configuring the Content Server to use the existing repository.
• Upgrading the 32-bit Content Server and repository to 64-bit Content Server 7.x.

To understand the migration process, consider the following scenario as a typical deployment of a pre-7.x 32-bit Content Server in your environment.

Host 1 is running a 32-bit operating system on which a 32-bit pre-7.x Content Server is installed. Host 2 is the 64-bit database server running on a 64-bit operating system. In case you have a 32-bit database server, you must upgrade it to 64-bit. The 32-bit database client libraries are installed on
Host 1 and are configured to point to the database server on Host 2. The following diagram illustrates the described environment.

**Figure 5. Pre-7.x 32-bit Content Server Environment**

You want to upgrade the 32-bit pre-7.x Content Server on 32-bit operating system to 64-bit Content Server 7.x. Since Content Server 7.x requires a 64-bit operating system to run on and because no direct upgrade of a 32-bit operating system to its 64-bit version is supported, a migration of the repository is required before you can upgrade.

To migrate the repository, create another system, Host 3, which runs a 64-bit operating system that is compatible with the Content Server on Host 1. The 32-bit database client libraries are also installed on Host 3 and configured to point to Host 2. Install the same 32-bit Content Server that is running on Host 1 on Host 3. Perform a migration of the repository filestores and content from Host 1 to Host 3.

After completing all the migration tasks, configure the Content Server to use the existing repository. Upgrade the operating system on Host 3 to the 64-bit supported version for Content Server 7.x, as specified in the *EMC Documentum Environment and System Requirements Guide*. Upgrade the database on Host 2 to the supported version. Upgrade the database client libraries to the supported 64-bit version. Finally, upgrade the 32-bit Content Server and repository to 64-bit Content Server 7.x.

The following diagram illustrates the entire migration process.
Migration Checklist

Perform the following tasks for migrating Content Server.

Table 17. Migration Checklist

<table>
<thead>
<tr>
<th>Step</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Prepare a plan for your migration.</td>
<td>See Planning a Migration, page 80.</td>
</tr>
<tr>
<td>2. Back up your repository.</td>
<td>Several third-party tools are available for backup.</td>
</tr>
<tr>
<td>3. Clean up your repository.</td>
<td>The “Repository maintenance” section in the EMC Documentum Content Server Administration and Configuration Guide provides more information.</td>
</tr>
<tr>
<td>4. Run the Consistency Checker utility.</td>
<td>The Upgrading the Content Server Software, page 40 provides the steps.</td>
</tr>
<tr>
<td>5. Fix any errors identified by the Consistency Checker.</td>
<td></td>
</tr>
<tr>
<td>6. Back up your cleaned, consistent repository.</td>
<td>Several third-party tools are available for backup.</td>
</tr>
</tbody>
</table>
7. Ensure that filestore_01 is online.

You can check this two ways:

1. Using Documentum Administrator, go to the Storage node and verify that filestore_01 shows as online.

2. Dump the filestore and check the r_status attribute.

   API>retrieve,c, dm_filestore where name='filestore_01'  
   API>dump,c,1

   Valid values are: 0, for on-line; 1, for off-line; 2, for read-only.

8. Install the 32-bit pre-7.x Content Server on the target host.

The EMC Documentum Content Server Installation Guide provides the installation steps.

9. Migrate the repository.

See Migration Methods, page 81.

10. Configure Content Server to use your existing repository.

The EMC Documentum Content Server Installation Guide provides the steps.

11. Upgrade Content Server and repository to 7.x.

The Upgrading the Content Server Software, page 40 provides the upgrade steps.

---

### Planning a Migration

Before you create the repository copy, complete these tasks and note any appropriate values in the Value column:

#### Table 18. Premigration Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Resource</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decide whether to copy the content files.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obtain the repository name.</td>
<td>Consult the repository administrator.</td>
<td>repository name: _________</td>
</tr>
<tr>
<td>Obtain the repository ID from the server.ini file.</td>
<td>Consult the repository administrator.</td>
<td>repository ID: _________</td>
</tr>
<tr>
<td>Obtain the repository owner’s name and password.</td>
<td>Consult the repository administrator.</td>
<td>repository owner: _________</td>
</tr>
<tr>
<td>Create a database instance separate from the database instance used by the production repository.</td>
<td>Consult the DBA.</td>
<td></td>
</tr>
</tbody>
</table>
Obtain connection information for the alternative database instance. | Consult the DBA and database documentation | connection: __________
---|---|---
Identify a target host on which to create the repository copy. | target host: __________
---|---|---
Obtain the system or administrator username and password for the database. | Consult the DBA. | admin user name: __________
---|---|---
Note the drive on which the production repository resides. | drive: __________
---|---|---
Decide whether to create the copy on the equivalent drive.
---|---|---
Decide whether to create the copy on a drive equivalent to the drive on which the production repository resides. If the copy is on a different drive, there are additional steps you must perform.
---|---|---

**Migration Methods**

When migration a repository, you can use the following two methods depending on the type of environment:

- Migrating a repository
- Copying a repository

If the target operating system supports upgrade from source operating system, use the *Copying a repository* method. For example, Content Server 7.0 on 64-bit Windows Server 2008 R2 to Content Server 7.1 on 64-bit Windows Server 2012.

If the target operating system does not support upgrade, use the *Migrating a repository* method. For example, Content Server 6.5 on 32-bit Windows Server 2003 R2 to Content Server 7.1 on 64-bit Windows Server 2012.

In the instructions that follow, the target repository host is called the *target host*. The source repository is called the *production repository*.

⚠️ **Caution:** The instructions that follow assume that the production repository is running on the network while the target host is tested. However, shut down the production repository or take it off the network while you test the target host. Conflicts and data corruption can result from having two repositories on the network with the same name and repository ID.
Method 1: Migrating a Repository

To migrate a repository:

1. On the target host, install the 64-bit version of the operating system and the 32-bit version of the database client libraries.

2. Create a new Content Server installation and repository (the repository copy) of the same version number as the production repository. The EMC Documentum Content Server Installation Guide provides the installation steps.

3. Copy $DOCUMENTUM/dba/secure/aek.key and $DOCUMENTUM/data from the production repository host to the same locations on the target host.

4. Create a new connection broker on the target host using the Content Server Configuration Program. From $DM_HOME/install, double-click Server_Configuration_Program.exe to run the Content Server Configuration Program.

5. Configure the repository.
   - When you create the new repository, ensure that you use the same repository name, repository ID, and repository owner name and password as the production repository.
   - Ensure that you use the same database instance used by the production repository. In the installer, ensure that you select the option to use an existing database user account.
   - If you are using LDAP for user authentication, copy the ldap*.cnt files from the original host to the repository copy host. In addition, copy the LDAP certificate database to the repository copy host.

6. Copy the $DOCUMENTUM/dba/config/<docbase-name>/dbpasswd.txt file from production repository host to the same location on the target host after providing the XML Store details.

7. Ensure that the repository configuration completes successfully.

8. Retrieve all the dm_jms_config objects from the repository copy and verify that the base_uri attribute and projection_targets attribute are set correctly.
   a. Start IDQL and connect to the repository as the installation owner.
   b. Run the following command to get a list of all dm_jms_config objects:
      ```
      select * from dm_jms_config
      ```
   c. Run the following command to change each projection target:
      ```
      update dm_jms_config objects set projection_targets[x]='new_target'
      where r_object_id='object_id'
      ```
   d. Run the following command to change each base URI:
      ```
      update dm_jms_config objects set base_uri[x]='new_base_uri'
      where r_object_id='object_id'
      ```

9. If you are testing the migration of a Web content management repository, modify the user objects to reflect the new authentication domain.
   a. Start IDQL and connect to the repository as the installation owner.
   b. Run the following commands:
      ```
      update dm_user objects
      ```
set user_os_domain = 'new_machine_name' where user_os_domain = 'old_machine_name'
c. Disconnect from the repository and exit IDQL.

10. If Content Server and content files of the copy reside on a drive different from the drive used by the production repository, use IDQL to update the file_system_path attribute of the dm_location and dm_mount_point objects to the new location:

   update dm_location objects
   set file_system_path = 'newpath' where file_system_path = 'old path'
   update dm_mount_point objects
   set file_system_path = 'newpath' where file_system_path = 'old path'

11. Restart Content Server.

**Method 2: Copying a Repository**

To copy a repository:

1. Shut down the production repository.

2. On the target host, create a new Content Server installation and repository (the repository copy) of the same version number as the production repository. The *EMC Documentum Content Server Installation Guide* provides the installation steps.

   • When you create the repository copy, ensure that you use the same repository name, repository ID, and repository owner name and password as the production repository.

   • Ensure that you use a different database instance from the instance used by the production repository and that you provide the correct connection information when you install.

      For example, under Oracle the *tnsnames.ora* on the host where the repository copy resides must point to the Oracle instance used by the copy, not the instance used by the production repository.

   • Ensure that the repository copy projects to a connection broker different from the connection broker used by the production repository.

   • **Copy the** $DOCUMENTUM/dba/secure/aek.key **and** $DOCUMENTUM/dba/secure/dbpasswd.txt **files from the original host to the same location on the repository copy host.**

   • If you are using LDAP for user authentication, copy the ldap*.cnt files from the original host to the repository copy host.

3. Apply to the repository copy any patches you applied to the production repository.

4. Connect to the database instance serving the production repository.

5. Use the database vendor’s tools to export all objects owned by the repository owner and export the schema for the tables comprising the repository.

   Contact the database vendor for any technical support you would need to use the database tools.

6. On the production repository host’s file system, create a backup of the $DOCUMENTUM/data/repository_name directory. This is the directory containing the repository’s content files.

7. Stop the repository copy.
8. Connect as the database system administrator to the database instance that is serving the repository copy. For example, on Oracle, connect as the System account.

9. Destroy the existing tablespaces or database by using the `dm_DeleteTableSpace.sql` script in `$DOCUMENTUM/dba/config/repository_name/`. The scripts are database-specific. Run the script using the tools provided by the database vendor.

10. Delete the physical database file from the file system. The name and location of the physical file are in the `dm_CreateTableSpace.sql` script.

11. Create new tablespaces or databases for the repository copy by using the `dm_CreateTableSpace.sql` script in `$DOCUMENTUM/dba/config/repository_name/`. The scripts are database-specific. Run the script using the tools provided by the database vendor.

12. Import the database export taken from the production repository into the newly created tablespaces or database.

13. Verify that the database tables have the correct value for the test system host name by checking the following values:
   - `r_host_name` and `web_server_loc` in `dm_server_config_s`
   - `host_name` in `dm_mount_point_s`
   - `target_server` in `dm_job_s`
   - `projection_targets` in `dm_server_config_r`
   - `object_name` from `dm_sysobject_s` where `r_object_type='dm_acs_config'`
   - `acs_base_url` in `dm_acs_config_r`

14. Connect to the database that is serving the repository copy as the repository owner.

15. If any of the values in Step 13 are incorrect, use SQL Server to correct the values.

16. Set the server to rebuild the Documentum views with this SQL Server statement:

   ```sql
   update dm_type_s set views_valid=0
   ```

17. If you are testing operations that require the content files, copy the content file backup from the production repository to the file system of the repository copy.

18. Navigate to the `DOCUMENTUM/dba/config/repository_name` directory and open the `server.ini` file in a text editor.

19. Ensure that the `preserve_existing_types` key in the `SERVER_STARTUP` section is set to `TRUE`:

   ```ini
   preserve_existing_types=T
   ```

20. Ensure that the crypto configuration parameters are set in the `SERVER_STARTUP` section.

   - For the repository created on pre 7.0 version, add the following settings to `server.ini` file:

     ```ini
     # RKM configuration parameters
     crypto_mode = 3DES_RSA1024_SHA256
     crypto_keystore = Local
     ```
• For the repository created in 7.0 or 7.x versions, the settings present in server.ini file on the source machine must be copied to server.ini file of target machine.

21. Save the server.ini file.

22. Start Content Server for the repository copy.

23. Retrieve all the dm_jms_config objects from the repository copy and verify that the base_uri attribute and projection_targets attribute are set correctly.
   a. Start IDQL and connect to the repository as the installation owner.
   b. Run the following command to get a list of all dm_jms_config objects:
      
      ```
      select * from dm_jms_config
      ```
   c. Run the following command to change each projection target:
      
      ```
      update dm_jms_config objects set projection_targets[x]='new_target'
      where r_object_id='object_id'
      ```
   d. Run the following command to change each base URI:
      
      ```
      update dm_jms_config objects set base_uri[x]='new_base_uri'
      where r_object_id='object_id'
      ```
   e. Restart Content Server for the changes to take effect.

24. If you are testing the migration of a Web content management repository, modify the user objects to reflect the new authentication domain.
   a. Start IDQL and connect to the repository as the installation owner.
   b. Run the following commands:
      
      ```
      update dm_user objects
      set user_os_domain = 'new_machine_name'
      where user_os_domain = 'old_machine_name'
      ```
   c. Disconnect from the repository and exit IDQL.

25. If Content Server and content files of the copy reside on a drive different from the drive used by the production repository, use IDQL to update the file_system_path attribute of the dm_location and dm_mount_point objects to the new location:

      ```
      update dm_location objects
      set file_system_path='newpath' where file_system_path='old path'
      update dm_mount_point objects
      set file_system_path='newpath' where file_system_path='old path'
      ```

26. Deactivate all jobs by changing the is_inactive attribute on all job objects to TRUE.

**Migrating Data Using SQL Server**

When migrating data using SQL Server 2008, you can use SQL Server Management Studio to import data or to export data from a source database table to a destination table.

When migrating data using SQL Server 2005, you can use SQL Server Management Studio to import data or to export data from a source database table to a destination table. If you use this mechanism for importing or exporting data, you may find that the identity columns of the destination tables do
not contain the same identity values as the source tables. This is a known issue with the SQL Server Import and Export Wizard. For more information, refer to the Microsoft support article.

To resolve this issue, follow these steps:

1. In the Column Mappings dialog box, select the Enable identity insert option.
2. Under Mappings, select a source table.
3. Select Create destination table and then click Edit SQL.
4. In the Create Table SQL Statement dialog box, replace the CREATE TABLE SQL statement with a SQL statement that includes an IDENTITY clause as shown in the following example:

   ```sql
   CREATE TABLE [dbo].[stud](
       [roll] [smallint] IDENTITY(1,1) NOT NULL,
       [name] [nvarchar](50) NULL,
       CONSTRAINT [PK_stud] PRIMARY KEY CLUSTERED
       (
           [roll] ASC
       )
   )
   ON [PRIMARY]
   ```

   **Note:** Assuming that dbo is the schema name.

5. Click OK and then click OK.

6. Proceed with the remaining steps in the SQL Server Import and Export Wizard.

7. You may need to update the query optimization statistics and recreate indexes to improve the database performance by using the following SQL statements:

   ```sql
   EXEC sp_MSForEachTable 'Update Statistics? WITH FULLSCAN'
   EXEC sp_MSforeachtable @command1="print 'DBCC DBREINDEX (')'"
   ```


   **Note:** If you are using Windows Server 2008 R2 and see database performance issues, check the Power Options settings in the Control Panel. The default power plan setting of Balanced is not recommended for SQL Server because of performance issues. Instead, set the power plan option to High performance. For more information, refer to the Microsoft article, Degraded overall performance on Windows Server 2008 R2.

---

**Consolidating Repositories**

In this release, you can perform the consolidation of repositories for the same versions of the Content Server. Ensure that the version of Content Server is 7.2 with AEK key residing in the RSA lockbox.

In addition, ensure that the source and target machines have separate database instances of SQL Server. In this release, the consolidation of repositories has been tested for the Windows SQL Server only.

To consolidate repositories, you must configure the Content Server on both the source and target machines. On both the machines, you have to set non-default and unique values for repository
name and repository ID. In addition, set different and unique names for the AEK key and lockbox, other than the default values.

For example, on the source machine, set the path for the Documentum folder as
C:/DocumentumSource, repository name as repol, and repository ID as 12345.

Once the configuration of the Content Server is complete, perform the following steps to consolidate the repositories on both the machines:

1. To migrate the SQL Server backup to the target machine, take the backup of the SQL Server database from SQL Server Management Studio and copy it to the target machine.

2. Copy the backup database file from the source to the target machine. Navigate to the SQL server management studio application of the target machine. Right click database-restore database and select the location from where you have copied the backup file.

3. Create new tablespaces or databases for the repository copy by using the dm _CreateTableSpace.sql script in $DocumentumSource/dba/config/<repository_name>. The scripts are database-specific. Run the script by using the tools provided by the database vendor.

4. Ensure that you note down the following environment variables that are set in the source machine in a .bat file on the target machine.
   - ClassPath
   - dfcpath
   - DM_HOME
   - DOCUMENTUM
   - JAVA_HOME
   - path

   Once you create the .bat file on the target machine, ensure that you update the values of the environment variables as per the folders in the target host.

5. Copy the Documentum folder from the source host to the target host.

   For example, copy the C:\DocumentumSource folder on the source host to the C:\DocumentumTarget folder on the target host.

6. On the target host, change the name of the docbroker.ini.

   For example, change the name of docbroker.ini to docbrokermig.ini.

7. On the target host, change the name of the connection broker in the dm_documentum_config .txt file.

   For example, if you want to change the name of the connection broker to Docbroker_Mig, make the following change in the dm_documentum_config.txt file:
   
   [DOCBROKER_DocbrokerMig]
   NAME=DocbrokerMig

8. In the dfc.properties file, make the following changes:
   - Change the port of the connection broker from the default value to 1889.
   - Change the value of dfc.data.dir and dfc.tokenstorage.dir to the updated folder paths.
   - Change the host name of the connection broker to the host name of the target machine.
9. In the server.ini file, make the following changes:
   - Update the path of the dbpasswd.txt file.
   - Change the port of the connection broker from the default value to 1889.
   - Change the value of the user_auth_target variable to the host name of the target machine.
   - Change the value of the [DOC BROKER_PROJECTION_TARGET] host variable to the host name of the target.

10. Add the names of the connection broker and the repository in C:\Windows\System32\drivers\etc\services along with the modified port and the name of the connection broker.
    For example, if you have set the name of the connection broker as DocbrokerMig, then make changes as follows:
    
    ```
    DocbrokerMig  1889/tcp  #docbroker
    DocbrokerMig_s  1890/tcp  #docbroker for secure connect
    dm_reponame  49625/tcp
    dm_reponame_s  49626/tcp  #Documentum Docbase Servicerepo256aes
    ```

11. Modify the default value of the port from 908X to any other value so that there is no port conflict.
    Make this modification in the following files:
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\configuration\standalone.xml
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\configuration\dctm.properties
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\configuration\standalone_xml_history\standalone.boot.xml
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\configuration\standalone_xml_history\standalone.initial.xml
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\configuration\standalone_xml_history\standalone.last.xml
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\deployments\acs.ear\lib\configs.jar\jmx.properties
    - %DOCUMENTUM%\jboss7.1.1\server\startMethodServer.cmd
    - %DOCUMENTUM%\jboss7.1.1\server\stopMethodServer.cmd

12. Update the folder names and paths in the following files:
    - %DOCUMENTUM%\jboss7.1.1\server\serviceConfig\MethodServer\conf\wrapper.conf
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\deployments\ServerApps.ear\DmMethods.war\WEB-INF\web.xml
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\deployments\acs.ear\lib\configs.jar\dfc.properties
    - %DOCUMENTUM%\jboss7.1.1\server\DctmServer_MethodServer\deployments\acs.ear\lib\configs.jar\log4j.properties
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/XhiveConnector.ear/APP-INF/classes/dfc.properties
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/XhiveConnector.ear/APP-INF/classes/log4j.properties
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/ServerApps.ear/APP-INF/classes/dfc.properties
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/ServerApps.ear/APP-INF/classes/log4j.properties
• %DOCUMENTUM%/jboss7.1.1/bin/dctmServerStatus.bat

13. Update the folder name, port, and host name in the following files:
• %DOCUMENTUM%/jboss7.1.1/server/serviceConfig/MethodServer/conf/DmMethodServer.xml
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/acs.ear/lib/configs.jar/config/acs.properties
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/acs.ear/lib/configs.jar/config/acsfull.properties
• %DOCUMENTUM%/product/7.2/bin/xdb.properties
• %DOCUMENTUM%/xhive_storage/XhiveDatabase.bootstrap
• %DOCUMENTUM%/jboss7.1.1/server/DctmServer_MethodServer/deployments/XhiveConnector.ear/XhiveConnector.war/WEB-INF/web.xml

14. Update the following values in the database tables in the target host:
• dm_server_config_s in r_host_name
• dm_server_config_r in app_server_uri
• dm_acs_config_r in acs_base_url
• dm_jms_config_r in base_url
• dm_user_s in user_os_domain
• dm_user_s in user_login_domain
• dm_user_s in user_global_unique_id
• dm_mount_point_s in host_name
• dm_job_s in target_server

Note: Update the value of any job that has old hostname, method, or parameters.
• dm_client_rights_s in host_name
• dm_client_registration_s in host_name
• dmExtern_store_r in a_storage_param_value
• dmr_content_s in set_client
• dm_sysobject_s in r_lock_machine
• dm_audittrail_s in host_name
Migrating Content Server

- `dm_sysprocess_config_r in base_uri`
- `dm_sysprocess_config_r in projection_targets`
- `dm_location_s in file_system_path`
- `dm_mount_point_s in file_system_path`
- `dmr_content_s in set_file`
- `dm_method_s in method_verb`
- `dm_audittrail_s in attribute_list`
- `dm_audittrail_s in string_2`
- `dm_sysobject_s in subject`
- `dm_type_s in views_valid`

15. To start the connection broker, the repository, and the Java Method Server, perform the following step in the same command prompt where you executed the .bat file:

```
start %DM_HOME%\bin\dmdocbroker.exe -init_file %DOCUMENTUM%\dba\Docbroker1.ini -logfile %DOCUMENTUM%\dba\log\Docbroker1.Docbroker.log
```

```
start %DM_HOME%\bin\documentum.exe -docbase_name <repository_name> -security acl -init_file %DOCUMENTUM%\dba\config\<repository_name>\server.ini -install _owner Administrator -logfile %DOCUMENTUM%\dba\log\<repository_name>.log
```

16. If you have enabled the repository with the lockbox option, then it may not come up after the migration. To facilitate the repository with the lockbox enabled, run the following command:

```
dm_crypto_boot -keyname <AEK_key_name> -lockbox <lockbox_name>.lb -lockboxpassphrase <lockbox_passphrase> -passphrase <AEK_passphrase>
```

To verify, run the following command:

```
dm_crypto_create -check -keyname <AEK_key_name> -lockbox <lockbox_name>.lb -lockboxpassphrase <lockbox_passphrase> -passphrase <AEK_passphrase>
```

If you have used the custom paraphrase, then run the following command:

```
dm_crypto_boot -passphrase <AEK_passphrase> -all
```

Migrating Data from Earlier Versions of Content Server

There are no special requirements or considerations when migrating data from earlier versions of Content Server to 64-bit Content Server. Follow the standard procedures for your migration utility (for example, Crown Partners, Bluefish, FME).

Migrating XML Content to the XML Store

The *EMC Documentum XML Store 6.5 SP1 Migration Guide* provides information on migrating the XML content to the XML store in Content Server.
**Note:** Before migrating the XML content to an XML store, Content Server must be upgraded and migrated to Documentum 6.5 or later.

### 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 upgrading Content Server, run the Configurator tool to configure the internal Java Method Server service. The Configurator tool writes the location of the Java methods to the internal method server. The location of the methods directory is written to the web.xml file in the method server deployment directory, for example, C:\Documentum\JBoss_HOME\server\DctmServer\_MethodServer\deploy\ServerApps.ear\DmMethods.war\WEB-INF:

```xml
<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 (Linux/AIX/Solaris) 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 2 modules and DocApps do not need to be changed when you upgrade Content Server to Documentum 6.0 or later. If you want to change a DocApp or module on an upgraded Documentum 6.0 or later Content Server, create a project in Composer and add your BOF2 modules or DocApp. The EMC Documentum Composer 6.7 User Guide provides more information about working with modules and Documentum Archive (DAR) files.

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

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 generates a DAR file that can be installed in a new instance of Content Server or edited in place in an upgraded Content Server instance. The "Migrating DocApps" chapter in the EMC Documentum Composer 6.7 User Guide provides more information.
If you want your DFC 5.3 SP6 clients to use upgraded BOF2 modules, perform the following tasks:

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 SP6 rather than DFC 6.0 or later to ensure that they do not accidentally reference new interfaces.

To migrate custom Business Objects in an environment of 5.3 SP6 clients that access Content Server 6.0 or later, do the following:

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

- **Module or TBO**
  
  Make sure that your code works 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 6.0 or later.

### Post-Migration Tasks

Complete all migration-related tasks described in this chapter. Review and complete the tasks described in the following chapters, if necessary:

- **Chapter 6, Migrating DFC Customizations**
- **Chapter 7, Migrating DFS Customizations**
- **Chapter 8, Migrating CMIS Customizations**
- **Appendix A, Migrating DMCL APIs to DFC**

After completing the migration of the repository, you need to upgrade the repository and Content Server to 7.x. The *Upgrading the Content Server Software, page 40* provides the steps for upgrading Content Server.
Chapter 6

Migrating DFC Customizations

The Documentum Java-Com Bridge (DJCB) and Primary Interop Assembly (PIA) are deprecated from version 6.

The following topics describe how to migrate DFC customizations to version 6.5 SP2 or later:

- Java Class Changes, page 93
- Configuring DFC for Native IPv4 Operations, page 94
- Configuring 6.7 Clients to Work with Content Server 7.x, page 94
- Migrating Customizations to Business Objects, page 94
- Migrating DMCL API Calls to DFC API Calls, page 95
- Search Service, page 95
- Full Format Specifications No Longer Accepted, page 95
- Character String Handling Improved, page 95
- Aspects, a New BOF Module Type for Developers, page 96
- JMX Management of DfPreferences and dfc.properties, page 96
- DFC Deployment, page 96
- Configuration for AAC Tokens, page 97
- Setting the Maximum Number of Results Per Source, page 97
- DFC Does Not Support Linked Store Storage Areas, page 97
- External Storage, page 97
- DFC Does Not Support Optical Storage Devices, page 98

Java Class Changes

New classes, methods, and class members, as well as changed or deprecated methods and members, are documented in diff files available on EMC Online Support with the current migration guide. There are diff sets comparing DFC classes of versions 5.3 SP6 to 6.0 and 6.0 SPx to 6.5 SP2 and later.
Configuring DFC for Native IPv4 Operations

Since: version 6.5

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

• Specify a host with an IPv4 address in the dfc.properties file as the value of \( \text{dfc.docbroker.host} \).
• Disable the dual-stack operation for JVM.

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

Configuring 6.7 Clients to Work with Content Server 7.x

If you install the version of 6.7 SP2 for the clients such as TaskSpace to work with a fresh Content Server 7.x, you must perform the following steps to update the dm_bof_registry user password since Content Server 7.x uses a different encryption algorithm (FIPS/SHA1) than that used by earlier versions of Content Server (MD5):

1. Decrypt the dm_bof_registry user password using the decrypt API of the RegistryPasswordUtils class in DFC 7.x to get the plain text password.
2. Encrypt the plain text password using the encrypt API of the RegistryPasswordUtils class in DFC 6.7SP2 on the client.
3. Place the encrypted password in dfc.properties on the client.

Migrating Customizations to Business Objects

Since: version 6

The Business Object Framework (BOF) provides a framework for your customizations that can be accessed from various client applications and service-based architecture. The following kinds of DFC customizations should be migrated to Business Objects:

• Core custom action execution logic
• Process automation, for example, creating renditions during checkin, creating workflows after checkin
• Custom data handlers
• Helper methods in utility classes, for example, attaching or detaching a lifecycle, promoting or demoting a document
• Business validation, for example, permitting an export operation
Examples of BOF classes

Updating Attributes of an Object Based on its Location — Generally, you organize documents in a meaningful folder hierarchy. You can also set one or more attributes on an object based on the location in which it is imported or created. The BOF module contains a TBO that sets the attribute after the operation, based on the parent folder.

Attaching a Lifecycle During a Checkin Operation — An SBO can be used to perform an operation after checkin, such as attaching a lifecycle. Other possible operations include promoting a workflow or creating a rendition.

Migrating DMCL API Calls to DFC API Calls

Since: version 6

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

The Appendix A, Migrating DMCL APIs to DFC section provides a map of DMCL APIs to DFC APIs.

Search Service

The DFC search service replaces prior mechanisms for building and running queries. You can use the IDifQuery interface, which is not part of the search service, for simple queries.

The search service provides the ability to run searches across multiple Documentum repositories and external repositories as well. The Javadocs for the com.documentum.fc.client.search package describe how to use this capability.

Full Format Specifications No Longer Accepted

Since: version 6

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

Character String Handling Improved

Since: version 6

In previous releases, if you attempted to set a character string property with a value that exceeded the defined length of the property, DFC silently truncated the value to the maximum length of the
property, then set the property. For Documentum 6, DFC throws an exception instead of truncating the value and setting the property.

To use the pre-Documentum 6 behavior, set the `dfc.compatibility.truncate_long_values` property in the `dfc.properties` file to `T`. This property is `False` by default.

### Aspects, a New BOF Module Type for Developers

**Since: version 6**

Documentum 6 supports aspects, a new framework for extending object behavior and attributes. Aspects are a type of BOF entity that can be dynamically attached to object instances in order to provide fields and methods beyond the standard ones for the object type. The extended behavior can include functionality that applies to types across the object hierarchy. For example, an aspect could label objects as retainable or web-viewable, and this single aspect could be applied to multiple distinct object types.

Aspects can speed development and improve code reuse, because the extended attributes and behavior do not alter the underlying type definitions. You can create aspects and associate them with an individual object or an object type. If you associate them with an object type, the aspect is automatically associated with each new object of the specified object type. Aspects can also have properties defined for them. Properties defined for an aspect appear to users as if they are defined for the object type of the object to which the aspect is attached.

### JMX Management of DfPreferences and dfc.properties

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

### DFC Deployment

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

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

Setting the Maximum Number of Results Per Source

Administrators can enhance performance by adjusting the maximum number of results returned per source as the result of a query. The default value is 350. The maximum number of results to retrieve is set in the `dfc.properties` file using the parameter `dfc.search.max_results_per_source` as follows:

```
dfc.search.max_results_per_source=number_of_results
```

For example:

```
dfc.search.max_results_per_source=350
```

DFC Does Not Support Linked Store Storage Areas

Since version 6.5, DFC does 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 plug-in is configured to execute on the client host, reconfigure the plug-in to execute on the server. In 6.5 and later versions, DFC does not support executing the plug-in on the client. To configure the plug-in to execute on the server, set the `a_exec_mode` property of the storage object to `F` (false). The storage object is one of `dmExternFile`, `dmExternFree`, or `dmExternUrl`, 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.
Chapter 7

Migrating DFS Customizations

This chapter covers operations you must perform when migrating DFS customizations to DFS 7.x. It also includes functionality and compatibility changes that you must consider after migrating to DFS 7.x.

This chapter covers the following topics:

- Upgrading the DFS .NET Productivity Layer, page 99
- Restoring Trusted Certificates after Upgrading UCF, page 101
- Trusted Login is Disabled By Default, page 102
- Cookie Consistency Check, page 102
- .NET Framework Update, page 102

Upgrading the DFS .NET Productivity Layer

The following table provides you with an overview of the supported upgrade paths and the corresponding configurations needed for the upgrade.

Table 19. DFS .NET Productivity Layer Upgrade Matrix

<table>
<thead>
<tr>
<th>Original Version</th>
<th>New Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.6 Pn</td>
</tr>
<tr>
<td>6.6 Pm</td>
<td>B</td>
</tr>
<tr>
<td>6.7 SPx Pm</td>
<td>B</td>
</tr>
<tr>
<td>7.0 Pm</td>
<td></td>
</tr>
<tr>
<td>7.1 Pm</td>
<td></td>
</tr>
</tbody>
</table>

SP = Service Pack; SPx, SPy = Service Pack 1, Service Pack 2, and so on, (x < y)
P = Patch; Pm, Pn = Patch 1, Patch 2, and so on, (m < n)

Option A: recompile – recompile the DFS-based application.

Option B: xcopy – replace old versions with new ones without recompile.

Option C: publisher policy – install new versions to global assembly cache (GAC) and make both versions coexist.
Option D: application/machine configuration – configure the application configuration file or machine configuration file to redirect assemblies. The Microsoft article, Redirecting Assembly Versions provides more information.

Detailed information is covered in the following topics:
- Upgrading from a Version Earlier Than 7.x, page 100
- Upgrading from a Pre–7.x Minor Version or Service Pack, page 101

The Microsoft article, Deploying Assemblies provides more information about .NET deployment.

**Upgrading from a Version Earlier Than 7.x**

To upgrade from a version earlier than 7.x, you install the DFS 7.x productivity layer assemblies to the GAC. You can upgrade from these versions:
- 6.5 SP2
- 6.6
- 6.7 (or later minor version or SP)

**To upgrade:**

1. Deploy the new DFS 7.x productivity layer assemblies in one of the following ways:
   - If a working directory does not exist, then install it to the GAC.
     
     Note: Administrator privileges are required to perform GAC operations.
   - If a working directory does exist, then use the xcopy command to copy it to the working directory.

2. Deploy the publisher policy assemblies as follows:
   a. Choose the publisher policy assembly from the DFS SDK with appropriate version.
      For example, if you are upgrading from 6.6 to 7.x, choose the assembly \Policy.6.6.Emc.Documentum.FS.XXX.DLL from the SDK in the emc-dfs-sdk-7.x \lib\dotnet\PublisherPolicy directory.
   b. Install the publisher policy assembly to the GAC.
      The Microsoft article, Installing an Assembly into the Global Assembly Cache provides more information about installing assemblies to the GAC.

Once you have installed the assemblies to the GAC, they are shared by all of the applications on the machine. If you have multiple DFS-based applications on the same machine and you do not want to upgrade DFS for some of them, you can bypass the publisher policy assemblies in those applications by modifying the application configuration file. The Microsoft article, Redirecting Assembly Versions provides more information about bypassing the publisher policy assemblies.

To uninstall the upgraded assemblies from the GAC, you simply delete the assemblies from the GAC. The Microsoft article, Removing an Assembly from the Global Assembly Cache provides information about deleting assemblies from the GAC.
Upgrading from a Pre–7.x Minor Version or Service Pack

To upgrade, use the `xcopy` command to replace your existing DFS .NET productivity layer assemblies in your DFS application’s working directory with the new DFS SDK assemblies.

Restoring Trusted Certificates after Upgrading UCF

In DFS 7.x, UCF expects that JRE 7 is installed on the client. If JRE 7 is not found on the client, UCF automatically upgrades JRE to version 7. In some SSL environments, the UCF client may have imported some trusted certificates to the UCF JRE’s cacerts store before the upgrading. These trusted certificates will be lost during the JRE upgrading. Therefore, you have to import the trusted certificates into the cacerts store of the upgraded JRE. To do this, perform the following steps:

1. Encrypt the trust/key store password:
   a. Navigate to the following directory:
      
      ```
      Profile\Documentum\ucf\{HOSTNAME}\shared\bin\7.x.0000.<minor_version>
      ```
   
   b. Run following command in the console:
      
      ```
      java -cp ".\ucf-client-api.jar;\ucf-client-impl.jar"
      com.documentum.ucf.common.util.spi.BaseCipher <trust/key store password>
      ```
      
      You will see the output that resembles the following:
      
      ```
      cipher.name: ${cypher.name} cipher.secret.key: 
      ${[cipher.secret.key]} cipher.secret.key.algorithm:
      ${[cipher.secret.key.algorithm]} Encrypted password
      (e.g. https.truststore.password): ${[https.keystore/truststore.password]}
      Password encoding (e.g. https.truststore.password.encoding):
      {https.keystore/truststore.password.encoding}
      ```

2. Configure the UCF client:
   a. Locate and open the following configuration file:
      
      ```
      ${User Profile}\Documentum\ucf\{HOSTNAME}\shared\config\ucf.client.config.xml
      ```
   
   b. Add following options, and then save the file.
      
      ```
      <configuration name="com.documentum.ucf">
      ....  
      </configuration>
      ```
<value>${cipher.secret.key}</value>
</option>
<option name="cipher.secret.key.algorithm">
<value>${cipher.secret.key.algorithm}</value>
</option>
<option name="https.truststore.file">
<value>${https.truststore.file}</value>
</option>
<option name="https.truststore.password">
<value>${https.truststore.password}</value>
</option>
<option name="https.truststore.password.encoding">
<value>${https.keystore/truststore.password.encoding}</value>
</option>
</configuration>
</configurations>

**Trusted Login is Disabled By Default**

DFS 7.x enhances the trusted login mechanism for DFS server so that it is disabled by default. Because of this change, users who do not provide the correct password cannot access DFS services in default settings. DFS server enables trusted login only if you explicitly enable it by setting the `dfc.session.allow_trusted_login` property to `true` in the `dfc.properties` file.

**Cookie Consistency Check**

Previously, the DFS client enforced the check of cookie consistency. In DFS 7.x, the DFS server enforces check of cookie consistency.

**.NET Framework Update**

In previous releases, UCF .NET depends on the availability of .NET Framework 3.5 SP1 on the client machine on which the UCF assembly files are downloaded. Starting from DFS 7.x, the client machine must have .NET Framework 4.0 installed to support the .NET UCF integration.
Chapter 8

Migrating CMIS Customizations

This chapter covers operations you must perform and some functionality and compatibility changes that you must note after migrating to CMIS 7.x.

This chapter covers the following topic:

• getFolderParent Returns Feed, page 103

getFolderParent Returns Feed

The return type of the getFolderParent method is changed to Feed.

Previously, the getFolderParent method returned entries. To unify the returns of getFolderParent and getOjectParents, the getFolderParent method now returns feeds. As a result of this change, you have to modify your code for the applications that use the getFolderParent method.
Chapter 9

Migration Scenarios

This chapter describes some of the supported scenarios for migrating and upgrading a previous version of Content Server to 7.2.

Note: Although there can be multiple migration scenarios depending on the operating system/database combination, it is not possible to document all of those scenarios. This chapter only covers some of those scenarios that were tested. However, for a particular operating system/database combination, the migration steps do not vary much across Content Server versions.

This chapter contains the following topics:
- Migrating Content Server 6.5 SP2 to 7.2 — Windows/SQL Server, page 105

Migrating Content Server 6.5 SP2 to 7.2 — Windows/SQL Server

This migration scenario provides the step-by-step instructions for migrating and upgrading the 32-bit Content Server 6.5 SP2 installed on the Windows 2008 SP2 (32-bit) operating system and using SQL Server 2008 SP2 (64-bit) as the database with SQL Server 2008 SP2 (32-bit) client. The migration process involves:

1. Installing 32-bit Content Server 6.5 SP2 on a target host running Windows 2008 SP2 (64-bit) operating system.
2. Migrating the repository from the source host to the target host.
3. Upgrading the operating system on the target host to Windows Server 2012 (64-bit).
4. Upgrading the database to SQL Server 2012 SP1.
5. Upgrading the Content Server to 7.1.

The following table lists the base and upgraded versions of the Content Server, operating system, and database that are supported in this migration scenario.

Table 20. Base and Upgraded OS/DB/CS Versions

<table>
<thead>
<tr>
<th></th>
<th>Base Version</th>
<th>Upgraded Version</th>
</tr>
</thead>
</table>
Migration Scenarios

<table>
<thead>
<tr>
<th>Database</th>
<th>SQL Server 2008 SP2 (64-bit) with 32-bit client libraries</th>
<th>SQL Server 2012 SP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Server</td>
<td>32-bit Content Server 6.6</td>
<td>64-bit Content Server 7.2</td>
</tr>
</tbody>
</table>

Premigration Tasks

1. Back up the repository. If the repository contains customized repository formats (dm_format objects), back up the customized formats.
2. Clean up the repository.
3. Run the Consistency Checker tool. The syntax is:
   
   ```
   dmbasic -fconsistency_checker.ebs -eEntry_Point -- <repository_name> <superuser> <password>
   ```
4. Fix the inconsistencies reported by the Consistency Checker tool as errors.
5. Back up your cleaned, consistent repository.
6. Ensure that filestore_01 is online using any one of the following methods:
   - Using Documentum Administrator, go to the Storage node and verify that filestore_01 shows as online.
   - Dump the filestore and check the r_status attribute:
     
     ```
     API>retrieve,c,dm_filestore where name='filestore_01' API>dump,c,l
     ```

Migration Tasks

1. On the target host, install Windows 2008 SP2 (64-bit) operating system and the 32-bit SQL Server 2008 SP1 database client libraries.
2. Install the 32-bit Content Server 6.6. The EMC Documentum Content Server 6.6 Installation Guide provides the instructions for installed Content Server.
3. Copy `%DOCUMENTUM\dba\secure\aek.key` and `%DOCUMENTUM\data` from the production repository host to the same locations on the target host.
4. Create a new connection broker on the target host using the Content Server Configuration Program.
   a. From `%DM_HOME%\install`, run `Server_Configuration_Program.exe`
   b. Type the password for the installation owner and click Next.
   c. Select Custom configuration and click Next.
   d. When you configure a repository, optionally select the check box to enable database partitioning. By default, database partitioning is disabled.
   e. Select the option to configure both connection broker and repository and click Next.
   f. Choose Create a new connection broker and click Next.
   g. Type a connection broker name (default: Docbroker) and the port number on which the connection broker listens, or accept the defaults. The default port is 1489. If you are using the default port number, ensure that the next port number (1490) is available for use because the connection broker requires that two ports be reserved.
h. Click **Automatic** to have the connection broker automatically start when the host starts, or click **Manual** for manual startup.

i. Select the mode in which the connection broker connects to the repository and click **Next**.

j. To continue with the server configuration, select the **Continue with server configuration** check box and click **Next**.

k. Select **Create a repository** and click **Next**.

5. Configure the repository.

   a. Select **Create a new repository** and click **Next**.

   b. To enable Content Storage Services, select the check box, type the license key, and click **Next**.

   c. To enable Retention Policy Services, select the check box, type the license key, and click **Next**.

   d. To enable Federated Records Services, select the check box, type the license key, and click **Next**.

   e. To enable Records Manager, select the check box, type the license, and click **Next**.

   f. To enable Physical Records Manager, select the check box, type the license, and click **Next**.

   g. Click **Next** to accept the default data directory location or browse for a different location.

   h. Click **Next** to accept the default share directory location or type a new location.

   i. Click **Next** to accept the default fully qualified domain name.

   j. To enable data partitioning, select the check box and click **Next**.

   k. When you provide the repository information, ensure that you use the same repository name, repository ID, and repository owner name and password as the production repository.

   l. Select the authentication domain. If you are using LDAP for user authentication, copy the `ldap*.cnt` files from the original host to the repository copy host. In addition, copy the LDAP certificate database to the repository copy host.

   m. Specify whether Content server starts automatically or manually and click **Next**.

   n. Select the option to use an existing database user account and storage, and click **Next**. You must use the same database instance used by the production repository.

   o. On SQL Server, select an ODBC data source.

   p. Type the username for an existing DB user, the DB user’s password, the DB administrator username and password, and then click **Next**.

   q. Choose the correct index tablespace or datafile name and click **Next**.

   r. Accept or modify the Content Server initialization values and click **Next**.

   s. Configure the data files or data devices.

   t. Provide the SMTP server information.

   u. Decide whether to designate the current repository as a global registry.

   v. Accept or modify the repository configuration scripts and click **Next**.

   w. Choose whether to restart Content Server to enable SSL client connections.
x. Specify the port that the XML Store should use and the directory where the XML Store should be created.

y. Click Finish.

6. Copy the `%DOCUMENTUM\dba\config\<docbase-name>\dbpasswd.txt` file from production repository host to the same location on the target host after providing the XML Store details.

7. Ensure that the repository configuration completes successfully.

8. Retrieve all the `dm_jms_config` objects from the repository copy and verify that the `base_uri` attribute and `projection_targets` attribute are set correctly.
   - a. Start IDQL and connect to the repository as the installation owner.
   - b. Run the following command to get a list of all `dm_jms_config` objects:
     
     ```
     select * from dm_jms_config
     ```
   - c. Run the following command to change each projection target:
     ```
     update dm_jms_config objects set projection_targets[x]='new_target'
     where r_object_id='object_id'
     ```
   - d. Run the following command to change each base URI:
     ```
     update dm_jms_config objects set base_uri[x]='new_base_uri'
     where r_object_id='object_id'
     ```

9. If you are testing the migration of a Web content management repository, modify the user objects to reflect the new authentication domain.
   - a. Start IDQL and connect to the repository as the installation owner.
   - b. Run the following commands:
     ```
     update dm_user objects
     set user_os_domain = 'new_machine_name' where user_os_domain = 'old_machine_name'
     ```
   - c. Disconnect from the repository and exit IDQL.

10. If Content Server and content files of the copy reside on a drive different from the drive used by the production repository, use IDQL to update the `file_system_path` attribute of the `dm_location` and `dm_mount_point` objects to the new location:
    ```
    update dm_location objects
    set file_system_path='newpath' where file_system_path='old path'
    ```
    ```
    update dm_mount_point objects
    set file_system_path='newpath' where file_system_path='old path'
    ```

11. Review the Chapter 5, Migrating Content Server and complete the remaining migration-related tasks.

12. Restart Content Server.
**Post-Migration Tasks**

If you want to move the 3DES AEK key to lockbox, then perform the following steps:

1. Run `Server_Configuration_Program.exe`. Select **Upgrade an existing repository**. Click Next.
2. Select **Keep AEK key unchanged**. Click Next.
3. Select **Enable Lockbox**. Specify the Lockbox file name and Lockbox passphrase. Click Next.
4. Complete the configuration of the repository.
5. Review the Post-Migration Tasks, page 92 section and complete the remaining migration-related tasks.

If you want to upgrade the 3DES AEK algorithm to AES 128/192/256 algorithm and use lockbox, then perform the following steps:

1. Run `Server_Configuration_Program.exe`. Select **Upgrade an existing repository**. Click Next.
2. Select **Upgrade AEK key**. Click Next.
3. Select **Create new or update existing AEK key**.
4. Select AEK algorithm as per your choice. Specify the information for AEK key name, AEK passphrase, Lockbox file name, and Lockbox passphrase. Select **Enable Lockbox**.
5. Complete the configuration of the repository.
6. Review the Post-Migration Tasks, page 92 section and complete the remaining migration-related tasks.

**Preupgrade Tasks**

Follow the steps provided in the Preupgrade Tasks, page 59 section.

**Upgrade Tasks**

1. Upgrade the operating system to Windows Server 2012 (64-bit).
2. Upgrade the database to SQL Server 2012 SP1.
3. Follow the steps 4 through 27 in the Upgrade Tasks, page 60 section.

**Post-Upgrade Tasks**

Follow the steps provided in the Post-Upgrade Tasks, page 62 section.
Appendix A

Migrating DMCL APIs to DFC

This chapter provides information that can help you migrate a DMCL-based application to a DFC application. Refer to the EMC Documentum Foundation Classes 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 continue to work in Documentum 6.5 or later. 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 Documentum 6.5 or later, DocBasic applications automatically access the new dmcl.dll, which passes instructions back and forth to DFC via an emulator.

C++ accesses DMCL through dynamic links. The applications continue to work, but they are working with, in essence, the 6.0 version of DMCL (with some bug fixes). C++ applications using the dmcl40.dll do not have access to methods or interfaces introduced in version 6.5.

Methods with no corresponding DFC method

The following methods are not implemented in DFC 6.5 and later:

- Listmessage
- Lpq
- Reset
- Unprint
Methods with corresponding DFC methods

Table 21, page 112 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 21. DMCL API methods and corresponding DFC methods

<table>
<thead>
<tr>
<th>DMCL API method</th>
<th>DFC correspondence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interface</td>
</tr>
<tr>
<td>Abort, for transactions</td>
<td>IDfSession</td>
</tr>
<tr>
<td></td>
<td>IDfSessionManager</td>
</tr>
<tr>
<td>Abort, for work flow</td>
<td>IDfWorkflow</td>
</tr>
<tr>
<td>Acquire</td>
<td>IDfWorkItem</td>
</tr>
<tr>
<td>Addesignature</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td>Addesignature</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td>Addactivity</td>
<td>IDfProcess</td>
</tr>
<tr>
<td>Addlink</td>
<td>IDfProcess</td>
</tr>
<tr>
<td>Addnote</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td></td>
<td>IDfPackage</td>
</tr>
<tr>
<td>Addpackage</td>
<td>IDfWorkflow</td>
</tr>
<tr>
<td></td>
<td>IDfWorkItem</td>
</tr>
<tr>
<td>Addpackageinfo</td>
<td>IDfActivity</td>
</tr>
<tr>
<td>Addport</td>
<td>IDfActivity</td>
</tr>
<tr>
<td>Addrendition</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Addroutecase</td>
<td>IDfActivity</td>
</tr>
<tr>
<td>Anyevents</td>
<td>IDfSession</td>
</tr>
<tr>
<td>Append</td>
<td>IDfTypedObject</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Appendcontent</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td>Appendfile</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td>Appendpart</td>
<td>IDfSysObject</td>
</tr>
<tr>
<td>Appendstate</td>
<td>IDfPolicy</td>
</tr>
<tr>
<td>Apply</td>
<td>IDfSession, IDfQuery</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Archive</td>
<td>IDfSession</td>
</tr>
<tr>
<td>Migrating DMCL APIs to DFC</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assemble</th>
<th>IDfSysObject</th>
<th>assemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume</td>
<td>IDfSession</td>
<td>assume</td>
</tr>
<tr>
<td>Attach</td>
<td>IDfSysObject</td>
<td>attachPolicy, detachPolicy</td>
</tr>
<tr>
<td>Audit</td>
<td>IDfAuditTrailManager</td>
<td>registerEventForType, registerEventForObject, registerEvents, registerEventsFromQuery, registerEventsInFolder</td>
</tr>
<tr>
<td>Authenticate</td>
<td>IDfClient, IDfSession, IDfSessionManager</td>
<td>authenticate</td>
</tr>
<tr>
<td>Begintran</td>
<td>IDfSession, IDfSessionManager</td>
<td>beginTrans, beginTransaction</td>
</tr>
<tr>
<td>Bindfile</td>
<td>IDfSysObject</td>
<td>bindFile</td>
</tr>
<tr>
<td>Branch</td>
<td>IDfSysObject</td>
<td>branch</td>
</tr>
<tr>
<td>Cachequery</td>
<td>IDfQuery</td>
<td>execute</td>
</tr>
<tr>
<td>Changepassword</td>
<td>IDfSession</td>
<td>changePassword</td>
</tr>
<tr>
<td>Checkin</td>
<td>IDfSysObject</td>
<td>checkin</td>
</tr>
<tr>
<td>Checkinapp</td>
<td>IDfSysObject</td>
<td>checkinEx</td>
</tr>
<tr>
<td>Checkout</td>
<td>IDfSysObject</td>
<td>checkout, checkoutEx</td>
</tr>
<tr>
<td>Close</td>
<td>IDfCollection</td>
<td>close</td>
</tr>
<tr>
<td>Commit</td>
<td>IDfSession, IDfSessionManager</td>
<td>commitTrans, commitTransaction</td>
</tr>
<tr>
<td>Complete</td>
<td>IDfWorkitem</td>
<td>complete, completeEx, completeEx2</td>
</tr>
<tr>
<td>Connect</td>
<td>IDfSessionManager, IDfClient</td>
<td>newSession</td>
</tr>
<tr>
<td>Count</td>
<td>IDfTypedObject</td>
<td>getAttrCount</td>
</tr>
<tr>
<td>Create</td>
<td>IDfSession</td>
<td>newObject, newObjectWithType</td>
</tr>
<tr>
<td>CreateAudit</td>
<td>IDfAuditTrailManager</td>
<td>createAudit</td>
</tr>
<tr>
<td>Datatype</td>
<td>IDfTypedObject</td>
<td>getAttrDataType</td>
</tr>
<tr>
<td>Delegate</td>
<td>IDfWorkitem</td>
<td>delegateTask</td>
</tr>
<tr>
<td>Demote</td>
<td>IDfSysObject</td>
<td>demote, scheduleDemote, cancelScheduleDemote</td>
</tr>
<tr>
<td>Dequeue</td>
<td>IDfSession</td>
<td>dequeue</td>
</tr>
<tr>
<td>Dereference</td>
<td>IDfReplica, IDfMirror</td>
<td>dereferenceReplica, dereferenceMirror</td>
</tr>
<tr>
<td>Method</td>
<td>Object Type</td>
<td>Method</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Describe</td>
<td>IDfSession</td>
<td>describe</td>
</tr>
<tr>
<td>Destroy</td>
<td>IDfPersistentObject</td>
<td>destroy</td>
</tr>
<tr>
<td>Disassemble</td>
<td>IDfSysObject</td>
<td>disassemble</td>
</tr>
<tr>
<td>Disconnect</td>
<td>IDfSession</td>
<td>disconnect (in IDfSession)</td>
</tr>
<tr>
<td></td>
<td>IDfSessionManager</td>
<td>release (in IDfSessionManager)</td>
</tr>
<tr>
<td>Dump</td>
<td>IDfTypedObject</td>
<td>dump</td>
</tr>
<tr>
<td>Dumpconnection</td>
<td>IDfSessionManager</td>
<td>Use getStatistics method in IDfSessionManager to return an IDfStatisticsManager object, which has the getDocbases and getSessions methods, which return information equivalent to that returned by Dumpconnection</td>
</tr>
<tr>
<td>Dumploginticket</td>
<td>IDfClient</td>
<td>encryptPassword</td>
</tr>
<tr>
<td>Encryptpass</td>
<td>IDfClient</td>
<td>encryptPassword</td>
</tr>
<tr>
<td>Execquery</td>
<td>IDfQuery</td>
<td>execute</td>
</tr>
<tr>
<td>Execsql</td>
<td>IDfWorkflow</td>
<td>execute</td>
</tr>
<tr>
<td>Fetch</td>
<td>IDfSession</td>
<td>getObject, getObjectWithCaching</td>
</tr>
<tr>
<td>Flush</td>
<td>IDfSession</td>
<td>flush</td>
</tr>
<tr>
<td>Flushcache</td>
<td>IDfSession</td>
<td>flushCache</td>
</tr>
<tr>
<td>Flushconnectpool</td>
<td>IDfSessionManager</td>
<td>clearIdentities</td>
</tr>
<tr>
<td>Freeze</td>
<td>IDfSysObject</td>
<td>freeze</td>
</tr>
<tr>
<td>Get</td>
<td>IDfTypedObject</td>
<td>getBoolean, getInt, getDouble, getId, getString, getTime, getValue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getRepeatingBoolean, getRepeatingInt, getRepeatingDouble, getRepeatingId, getRepeatingString, getRepeatingTime, getRepeatingValue</td>
</tr>
<tr>
<td>Getconnection</td>
<td>IDfSessionManager</td>
<td>newSession</td>
</tr>
<tr>
<td>Getcontent</td>
<td>IDfSysObject</td>
<td>getContent</td>
</tr>
<tr>
<td>Getdocbasemap</td>
<td>IDfDocbrokerClient</td>
<td>getDocbaseMap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getDocbaseMapFromSpecificDocbroker</td>
</tr>
<tr>
<td>Getdocbrokermap</td>
<td>IDfDocbrokerClient</td>
<td>getDocbrokerMap</td>
</tr>
<tr>
<td>Getevents</td>
<td>IDfSession</td>
<td>getEvents</td>
</tr>
<tr>
<td>Getfile</td>
<td>IDfSysObject</td>
<td>getFile, getFileEx, getFileEx2</td>
</tr>
<tr>
<td>Getlastcoll</td>
<td>IDfSession</td>
<td>getLastCollection</td>
</tr>
<tr>
<td>Getlogin</td>
<td>IDfSession</td>
<td>GetLoginTicket, getLoginTicketEx, getLoginTicketForUser</td>
</tr>
<tr>
<td>Function</td>
<td>Class</td>
<td>Method</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Getmessage</td>
<td>IDfSession</td>
<td>getMessage</td>
</tr>
<tr>
<td>Getpath</td>
<td>IDfSysObject</td>
<td>getPath, getPathEx, getPathEx2</td>
</tr>
<tr>
<td>Getservermap</td>
<td>IDfDocbrokerClient</td>
<td>getServerMap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getServerMapFromSpecificDocbroker</td>
</tr>
<tr>
<td>Grant</td>
<td>IDfSysObject</td>
<td>grant,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>see also grantPermit</td>
</tr>
<tr>
<td>Halt</td>
<td>IDfWorkflow</td>
<td>halt, haltEx, haltAll</td>
</tr>
<tr>
<td>Id</td>
<td>IDfSession</td>
<td>getIdByQualification (in IDfSession)</td>
</tr>
<tr>
<td></td>
<td>IDfTypedObject</td>
<td>getObjectId (in IDfTypedObject)</td>
</tr>
<tr>
<td>Initcrypto</td>
<td>IDfClient</td>
<td>initCrypto</td>
</tr>
<tr>
<td>Insert</td>
<td>IDfTypedObject</td>
<td>insertBoolean, insertInt, insertDouble,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>insertId, insertString, insertTime,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>insertValue</td>
</tr>
<tr>
<td>Insertcontent</td>
<td>IDfSysObject</td>
<td>insertContent, insertContentEx</td>
</tr>
<tr>
<td>Insertfile</td>
<td>IDfSysObject</td>
<td>insertFile, insertFileEx</td>
</tr>
<tr>
<td>Insertpart</td>
<td>IDfSysObject</td>
<td>insertPart</td>
</tr>
<tr>
<td>Insertstate</td>
<td>IDfPolicy</td>
<td>insertState</td>
</tr>
<tr>
<td>Install</td>
<td>IDfActivity, IDfPolicy, IDfProcess</td>
<td>install</td>
</tr>
<tr>
<td>Invalidate</td>
<td>IDfActivity, IDfPolicy, IDfProcess</td>
<td>invalidate</td>
</tr>
<tr>
<td>IScached</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kill</td>
<td>IDfSession</td>
<td>killSession (for sessions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flushObject (for SysObjects)</td>
</tr>
<tr>
<td>Link</td>
<td>IDfSysObject</td>
<td>link</td>
</tr>
<tr>
<td>Listconnection</td>
<td>IDfSessionManager</td>
<td>Use getStatistics method in IDfSessionManager to return an IDfStatisticsManager object, which has the getDocbases and getSessions methods, which return information equivalent to that returned by Listconnection</td>
</tr>
<tr>
<td>Locate</td>
<td>IDfTypedObject</td>
<td>findBoolean, findInt, findDouble, findId,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>findString, findTime, findValue</td>
</tr>
<tr>
<td>Lock</td>
<td>IDfPersistentObject</td>
<td>lock</td>
</tr>
<tr>
<td>Mark</td>
<td>IDfSysObject</td>
<td>mark</td>
</tr>
<tr>
<td>Mount</td>
<td>IDfSysObject</td>
<td>mount</td>
</tr>
<tr>
<td>Movestate</td>
<td>IDfPolicy</td>
<td>moveState</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td>IDfCollection</td>
<td></td>
</tr>
<tr>
<td>Offset</td>
<td>IDfTypedObject</td>
<td></td>
</tr>
<tr>
<td>Pause</td>
<td>IDfWorkitem</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td>Promote</td>
<td>IDfSysObject, promote, schedulePromote, cancelSchedulePromote</td>
<td></td>
</tr>
<tr>
<td>Prune</td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td>Publish_dd</td>
<td>IDfSession, publishDataDictionary</td>
<td></td>
</tr>
<tr>
<td>PurgeLocal</td>
<td>IDfSession, purgeLocalFiles</td>
<td></td>
</tr>
<tr>
<td>Query_cmd</td>
<td>IDfQuery, execute</td>
<td></td>
</tr>
<tr>
<td>Query</td>
<td>IDfQuery, execute</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td>Readquery</td>
<td>IDfQuery, execute</td>
<td></td>
</tr>
<tr>
<td>Refresh</td>
<td>IDfReplica, refreshReplica</td>
<td></td>
</tr>
<tr>
<td>Reinit</td>
<td>IDfSession, reinit</td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td>IDfTypedObject, remove</td>
<td></td>
</tr>
<tr>
<td>Removeactivity</td>
<td>IDfProcess, removeActivity</td>
<td></td>
</tr>
<tr>
<td>Removecontent</td>
<td>IDfSysObject, removeContent</td>
<td></td>
</tr>
<tr>
<td>Removelink</td>
<td>IDfProcess, removeLink</td>
<td></td>
</tr>
<tr>
<td>Removenote</td>
<td>IDfSysObject, removeNote</td>
<td></td>
</tr>
<tr>
<td>Removepackage</td>
<td>IDfWorkitem, removePackage</td>
<td></td>
</tr>
<tr>
<td>Removepackageinfo</td>
<td>IDfActivity, removePackageInfo</td>
<td></td>
</tr>
<tr>
<td>Removeport</td>
<td>IDfSysObject, removePart</td>
<td></td>
</tr>
<tr>
<td>Removerendition</td>
<td>IDfSysObject, removeRendition, removeRenditionEx, removeRenditionEx2</td>
<td></td>
</tr>
<tr>
<td>Removeroutcase</td>
<td>IDfActivity, removeRouteCase</td>
<td></td>
</tr>
<tr>
<td>Removestate</td>
<td>IDfActivity, removeState</td>
<td></td>
</tr>
<tr>
<td>Repeat</td>
<td>IDfWorkitem, repeat</td>
<td></td>
</tr>
<tr>
<td>Repeating</td>
<td>IDfTypedObject, isAttrRepeating</td>
<td></td>
</tr>
<tr>
<td>Migrating DMCL APIs to DFC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Resolvealias</strong></td>
<td>IDfSysObject</td>
<td>IDfSession</td>
</tr>
<tr>
<td><strong>Restart</strong></td>
<td>IDfSession</td>
<td>IDfWorkflow</td>
</tr>
<tr>
<td><strong>Restore</strong></td>
<td>IDfSession</td>
<td></td>
</tr>
<tr>
<td><strong>Resume</strong></td>
<td>for lifecycles: IDfSysObject</td>
<td>IDfWorkflow</td>
</tr>
<tr>
<td><strong>Retrieve</strong></td>
<td>IDfSession</td>
<td>IDfTypedObject</td>
</tr>
<tr>
<td><strong>Revert</strong></td>
<td>IDfPersistentObject</td>
<td></td>
</tr>
<tr>
<td><strong>Revoke</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Save</strong></td>
<td>IDfPersistentObject</td>
<td></td>
</tr>
<tr>
<td><strong>Saveasnew</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Seek</strong></td>
<td>IDfContentCollection</td>
<td></td>
</tr>
<tr>
<td><strong>Set</strong></td>
<td>IDfTypedObject</td>
<td></td>
</tr>
<tr>
<td><strong>Setbatchhint</strong></td>
<td>IDfSession</td>
<td></td>
</tr>
<tr>
<td><strong>Setcontent</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Setcontentattrs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Setdoc</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Setfile</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Setoutput</strong></td>
<td>IDfWorkitem</td>
<td></td>
</tr>
<tr>
<td><strong>Setpath</strong></td>
<td>IDfSysObject</td>
<td></td>
</tr>
<tr>
<td><strong>Setperformers</strong></td>
<td>IDfWorkflow</td>
<td></td>
</tr>
<tr>
<td><strong>Setpriority</strong></td>
<td>IDfWorkitem</td>
<td></td>
</tr>
<tr>
<td><strong>Setsupervisor</strong></td>
<td>IDfWorkflow</td>
<td></td>
</tr>
<tr>
<td><strong>Shutdown</strong></td>
<td>IDfSession</td>
<td></td>
</tr>
<tr>
<td><strong>Signoff</strong></td>
<td>IDfPersistentObject</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Class</td>
<td>Descriptions</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Suspend</td>
<td>IDfSysObject</td>
<td>suspend, scheduleSuspend, cancelScheduleSuspend</td>
</tr>
<tr>
<td>Trace</td>
<td>IDfSession</td>
<td>TraceDMCL</td>
</tr>
<tr>
<td>Truncate</td>
<td>IDfTypedObject</td>
<td>removeAll, truncate</td>
</tr>
<tr>
<td>Type</td>
<td>IDfSession</td>
<td>getTypeDescription</td>
</tr>
<tr>
<td>Unaudit</td>
<td>IDfAuditTrailManager</td>
<td>unRegisterEvent, unRegisterEventForType, unregisterEvents,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unRegisterEventsFromQuery, unRegisterEventsInFolder, unRegisterAllEvents</td>
</tr>
<tr>
<td>Unfreeze</td>
<td>IDfSysObject</td>
<td>unfreeze</td>
</tr>
<tr>
<td>Uninstall</td>
<td>IDfActivity, IDfPolicy, IDfProcess</td>
<td>uninstall</td>
</tr>
<tr>
<td>Unlink</td>
<td>IDfSysObject</td>
<td>unLink</td>
</tr>
<tr>
<td>Unlock</td>
<td>IDfSysObject</td>
<td>cancelCheckOut</td>
</tr>
<tr>
<td>Unmark</td>
<td>IDfSysObject</td>
<td>unMark</td>
</tr>
<tr>
<td>Unregister</td>
<td>IDfSysObject</td>
<td>unRegisterEvent</td>
</tr>
<tr>
<td>Updatepart</td>
<td>IDfSysObject</td>
<td>updatePart, updatePartEx</td>
</tr>
<tr>
<td>Useacl</td>
<td>IDfSysObject</td>
<td>useACL</td>
</tr>
<tr>
<td>Validate</td>
<td>IDfActivity, IDfPolicy, IDfProcess</td>
<td>validate, validateProcessAndActivities</td>
</tr>
<tr>
<td>Values</td>
<td>IDfTypedObject</td>
<td>getValueCount</td>
</tr>
<tr>
<td>Vdmpath</td>
<td>IDfObjectPath</td>
<td>getAccessPath, getAccessibleFolderIds</td>
</tr>
<tr>
<td>Vdmpathdql</td>
<td>IDfObjectPath</td>
<td>getAccessPath, getAccessibleFolderIds</td>
</tr>
<tr>
<td>Verifyaudit</td>
<td>IDfPersistentObject</td>
<td>verifyAudit</td>
</tr>
<tr>
<td>Verifyesignature</td>
<td>IDfSysObject</td>
<td>verifySignature</td>
</tr>
</tbody>
</table>
These tables describe types and properties that are new, changed, deprecated, or obsolete since Documentum 6.7 SP1.

**New object types**

*Table 22, page 119* lists the new object types in Documentum 7.x.

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Description</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmc_bpm_lsm</td>
<td>Models a synchronizing split-join block (LSM) within a workflow template.</td>
<td>• join_act</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• process_id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• start_act</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• step_act</td>
</tr>
<tr>
<td>dmc_mq_config</td>
<td>Defines the configuration object of a message queue. An instance of dmc_mq_config is created when a message queue is created.</td>
<td>• queue_name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• max_redeliveries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• retain_dead_message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• retain_period</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• expiration_interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• default_priority</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• delivery_timeout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• queue_users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• permissions</td>
</tr>
</tbody>
</table>
# Changed object types

Table 23, page 120 lists the properties that have their lengths extended in Documentum 7.x.

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Properties</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm_user</td>
<td>user_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>user_group_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>user_login_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 80 to 255.</td>
</tr>
<tr>
<td></td>
<td>acl_domain</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>user_admin</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>user_delegation</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>default_folder</td>
<td>char(450) for SQL Server, char(740) for Oracle</td>
<td>Length of the property has been extended.</td>
</tr>
<tr>
<td></td>
<td>user_ldap_dn</td>
<td>char(512)</td>
<td>Length of the property has been extended from 256 to 512.</td>
</tr>
<tr>
<td>dm_group</td>
<td>group_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>group_admin</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>group_global_unique_id</td>
<td>char(400)</td>
<td>Length of the property has been extended from 255 to 400.</td>
</tr>
<tr>
<td></td>
<td>users_names</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>255. Forms a composite index with r_object_id.</td>
</tr>
<tr>
<td>groups_names</td>
<td>char(255)</td>
<td></td>
<td>Length of the property has been extended from 32 to 255. Forms a composite index with r_object_id.</td>
</tr>
<tr>
<td>i_supergroups_names</td>
<td>char(255)</td>
<td></td>
<td>Length of the property has been extended from 32 to 255. Used in an index by itself.</td>
</tr>
<tr>
<td>i_nondyn_supergroups_names</td>
<td>char(255)</td>
<td></td>
<td>Length of the property has been extended from 32 to 255. Forms a composite index with r_object_id.</td>
</tr>
<tr>
<td>dm_sysobject</td>
<td>r_modifier</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Used in an index by itself.</td>
</tr>
<tr>
<td></td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Used in an index by itself.</td>
</tr>
<tr>
<td></td>
<td>group_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_lock_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>acl_domain</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Forms a composite index with acl_name.</td>
</tr>
<tr>
<td></td>
<td>r_creator_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dm_acl</td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Forms two composite indexes: one with (i_partition, object_name) and another with (object_name, r_object_id, i_partition).</td>
</tr>
<tr>
<td></td>
<td>r_accessor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmi_package</td>
<td>r_note_writer</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmi_queue_item</td>
<td>supervisor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>sent_by</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>dequeued_by</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Forms one composite index with (name, sign_off_user, task_state, priority, date_sent).</td>
</tr>
<tr>
<td></td>
<td>sign_off_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. Form two composite indexes: one with (name, dequeued_by, task_state, priority, date_sent) and another with (item_id, task_state, name, priority, date_sent, r_object_id).</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmi_registry</td>
<td>user_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dmi_wf_attachment</td>
<td>r_creator_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_alias_set</td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_audit_policy</td>
<td>accessor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_type</td>
<td>group_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>group_global_unique_id</td>
<td>char(400)</td>
<td>Length of the property has been extended from 255 to 400.</td>
</tr>
<tr>
<td>dmi_type_info</td>
<td>default_group</td>
<td>char(255)</td>
<td>Length of the property has been extended from 27 to 255.</td>
</tr>
<tr>
<td></td>
<td>acl_domain</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_ftindex_agent_config</td>
<td>queue_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 64 to 255.</td>
</tr>
<tr>
<td>dm_registered</td>
<td>table_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 64 to 255.</td>
</tr>
<tr>
<td>dm_activity</td>
<td>performer_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 66 to 255.</td>
</tr>
<tr>
<td>dm_workitem</td>
<td>r_performer_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_ext_performer</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dm_auditrail</td>
<td>user_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>acl_domain</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_auditrail acl</td>
<td>accessor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_auditrail group</td>
<td>group_admin</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>users_names</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>groups_names</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_category</td>
<td>category_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_category assign</td>
<td>modifier</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>pre_modifier</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_ci_config</td>
<td>auto_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>manual_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_client rights</td>
<td>allowed_roles</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_docbase config</td>
<td>a_bpaction_run_as</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dm_docset_run</td>
<td>run_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_message_address</td>
<td>user_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255. This type has been changed to registered table.</td>
</tr>
<tr>
<td>dm_partition_scheme</td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_qual_comp</td>
<td>valid_groups</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_reference</td>
<td>r_ref_creator</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>i_ref_acl_domain</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_router</td>
<td>supervisor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>task_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_task_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_sign_off_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_server_config</td>
<td>operator_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_install_owner</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_webc_config</td>
<td>notification_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>Type Name</td>
<td>Properties</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------</td>
<td>-----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>dm_webc_target</td>
<td>transfer_user</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dm_workflow</td>
<td>r_creator_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>supervisor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_last_performer</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>r_performers</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_completed_workflow</td>
<td>creator_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>supervisor_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_completed_workitem</td>
<td>performer_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_module</td>
<td>a_privilege_roles</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_wfsdrp_parent</td>
<td>performer_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_workqueue_policy</td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td>dmc_workqueue_doc_profile</td>
<td>owner_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
</tbody>
</table>
### Object Type and Property Changes

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Properties</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmc_workqueue_user_profile</td>
<td>user_name</td>
<td>char(255)</td>
<td>Length of the property has been extended from 32 to 255.</td>
</tr>
<tr>
<td></td>
<td>owner_name</td>
<td>char(255)</td>
<td></td>
</tr>
<tr>
<td>dmc_wq_user_skill</td>
<td>user_name</td>
<td>char(255)</td>
<td></td>
</tr>
</tbody>
</table>

**Changed object types with new properties**

Table 24, page 127 lists the new properties added to existing object types in Documentum 7.x.

**Table 24. New object properties**

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Properties</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm_ftquery_subscription</td>
<td>zone_value</td>
<td>integer</td>
</tr>
<tr>
<td>dm_user</td>
<td>root_log_dir</td>
<td>char(255)</td>
</tr>
<tr>
<td>dm_type</td>
<td>r_creation_date</td>
<td>time</td>
</tr>
<tr>
<td></td>
<td>r_modified_date</td>
<td>time</td>
</tr>
<tr>
<td>Dm_acs_config</td>
<td>dormancy_status</td>
<td>string(32)</td>
</tr>
<tr>
<td>Dm_activity</td>
<td>activity_urn</td>
<td>string(512)</td>
</tr>
<tr>
<td></td>
<td>lsm_id</td>
<td>ID</td>
</tr>
<tr>
<td>Dm_docbase_config</td>
<td>dormancy_status</td>
<td>string(32)</td>
</tr>
<tr>
<td></td>
<td>i_crypto_keys_expiry_date</td>
<td>time</td>
</tr>
<tr>
<td></td>
<td>i_expired_crypto_keys</td>
<td>string(255)</td>
</tr>
<tr>
<td></td>
<td>r_crypto_keystore</td>
<td>string(32)</td>
</tr>
<tr>
<td></td>
<td>r_crypto_mode</td>
<td>string(64)</td>
</tr>
<tr>
<td>Dm_ldap_config</td>
<td>group_tree_sync</td>
<td>Boolean</td>
</tr>
<tr>
<td>Dm_process</td>
<td>system_name</td>
<td>string(512)</td>
</tr>
<tr>
<td>Dmr_content</td>
<td>content_state</td>
<td>integer</td>
</tr>
<tr>
<td>Dmi_wf_timer</td>
<td>r_is_initialized</td>
<td>integer</td>
</tr>
<tr>
<td>Dmi_change_record</td>
<td>dormancy_change_count</td>
<td>integer</td>
</tr>
<tr>
<td>Dmi_workitem</td>
<td>a_control_instruction</td>
<td>string(32)</td>
</tr>
</tbody>
</table>
Object Type and Property Changes

The EMC Documentum Content Server System Object Reference Guide provides additional information about the new, changed, deprecated, and obsolete object types and properties.
These topics are included:

- Overview, page 129
- Changes to existing key names, page 129
- dmcl.ini key migration to dfc.properties, page 132
- Obsolete dmcl.ini and session configuration options, page 133
- Obsolete dfc.properties keys, page 135

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 129, 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 Documentum 7.x. Invalid entries do not generate an error, but have no effect on functionality.

<table>
<thead>
<tr>
<th>Old name</th>
<th>New name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfc.acs.avail.refresh.frequency</td>
<td>dfc.acs.avail.refresh_interval</td>
</tr>
<tr>
<td>dfc.acs.config.refresh.frequency</td>
<td>dfc.acs.config.refresh_interval</td>
</tr>
<tr>
<td>dfc.admin.ldif.file.charset</td>
<td>dfc.admin.ldif_file_charset</td>
</tr>
<tr>
<td>dfc.cacs.check.keep.number</td>
<td>dfc.bocs.check.keep_number</td>
</tr>
<tr>
<td>dfc.cache.append.name</td>
<td>dfc.bof.cache.append_name</td>
</tr>
<tr>
<td>Old name</td>
<td>New name</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>dfc.bof.cacheconsistency.interval</td>
<td>dfc.bof.cache.currency_check_interval</td>
</tr>
<tr>
<td>dfc.bof.registry.connect.attempt.interval</td>
<td>dfc.globalregistry.connect_attempt_interval</td>
</tr>
<tr>
<td>dfc.bof.registry.preload.enabled</td>
<td>dfc.bof.cache.enable_preload</td>
</tr>
<tr>
<td>dfc.bof.registry.password</td>
<td>dfc.globalregistry.password</td>
</tr>
<tr>
<td>dfc.bof.registry.repository</td>
<td>dfc.globalregistry.repository</td>
</tr>
<tr>
<td>dfc.bof.registry.username</td>
<td>dfc.globalregistry.username</td>
</tr>
<tr>
<td>dfc.cache.ddinfo.globalCacheSize</td>
<td>dfc.cache.ddinfo.size</td>
</tr>
<tr>
<td>dfc.cache.dir</td>
<td>dfc.cache_dir</td>
</tr>
<tr>
<td>client_cache_size</td>
<td>dfc.cache.object.size</td>
</tr>
<tr>
<td>dfc.cache.query.globalCacheSize</td>
<td>dfc.cache.query.size</td>
</tr>
<tr>
<td>dfc.core.truncate_long_values</td>
<td>dfc.compatibility.truncate_long_values</td>
</tr>
<tr>
<td>dfc.config.timeout</td>
<td>dfc.config.check_interval</td>
</tr>
<tr>
<td>dfc.checkout.dir</td>
<td>dfc.data.checkout_dir</td>
</tr>
<tr>
<td>dfc.data.dir</td>
<td>dfc.data.dir</td>
</tr>
<tr>
<td>dfc.docbase.max_deadlock_retries</td>
<td>dfc.session.max_deadlock_retries</td>
</tr>
<tr>
<td>dfc.docbase.max_error_retries</td>
<td>dfc.session.max_error_retries</td>
</tr>
<tr>
<td>dfc.exception.include_decoration</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.exception.include_id</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.export.dir</td>
<td>dfc.data.export_dir</td>
</tr>
<tr>
<td>dfc.housekeeping.cleanup.interval,</td>
<td>dfc.bof.cache.cleanup_interval</td>
</tr>
<tr>
<td>dfc.resources.cleanup_interval</td>
<td></td>
</tr>
<tr>
<td>dfc.max.vdm.children.flush.count</td>
<td>dfc.vdm.max_child_flush_count</td>
</tr>
<tr>
<td>dfc.recordInlineDescendants</td>
<td>dfc.xml.record_inline_descendants</td>
</tr>
<tr>
<td>dfc.registry.file</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.registry.mode</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.resources.diagnostics.enabled</td>
<td>dfc.diagnostics.resources.enable</td>
</tr>
<tr>
<td>dfc.search.docbase.brokers</td>
<td>dfc.search.docbase.broker_count</td>
</tr>
<tr>
<td>dfc.search.ecis.adapter.domain</td>
<td>dfc.search.externalsources.adapter.domain</td>
</tr>
<tr>
<td>dfc.search.ecis.broker_count, dfc.search.ecis</td>
<td>dfc.search.externalsources.broker_count</td>
</tr>
<tr>
<td>.brokers</td>
<td></td>
</tr>
<tr>
<td>dfc.search.ecis.enable</td>
<td>dfc.search.externalsources.enable</td>
</tr>
<tr>
<td>dfc.search.ecis.host</td>
<td>dfc.search.externalsources.host</td>
</tr>
<tr>
<td>dfc.search.ecis.password</td>
<td>dfc.search.externalsources.password</td>
</tr>
<tr>
<td>dfc.search.ecis.port</td>
<td>dfc.search.externalsources.port</td>
</tr>
<tr>
<td>dfc.search.ecis.request_timeout, dfc.search.ecis.access.timeout</td>
<td>dfc.search.externalsources.request_timeout</td>
</tr>
<tr>
<td>Old name</td>
<td>New name</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>dfc.search.ecis.rmi_name, dfc.search.ecis.name</td>
<td>dfc.search.external_sources.rmi_name</td>
</tr>
<tr>
<td>dfc.search.ecis.username, dfc.search.ecis.login</td>
<td>dfc.search.external_sources.username</td>
</tr>
<tr>
<td>dfc.search.formatcache.timeout</td>
<td>dfc.search.formatcache.refresh_interval</td>
</tr>
<tr>
<td>dfc.search.fulltext.enabled</td>
<td>dfc.search.fulltext.enable</td>
</tr>
<tr>
<td>dfc.search.sourcecache.timeout</td>
<td>dfc.search.sourcecache.refresh_interval</td>
</tr>
<tr>
<td>dfc.search.typecache.timeout</td>
<td>dfc.search.typecache.refresh_interval</td>
</tr>
<tr>
<td>None</td>
<td>dfc.search.matching_terms_computing.enable</td>
</tr>
<tr>
<td>dfc.session.dynamic_delay</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.session.pool.enabled, connect_pooling_enabled</td>
<td>dfc.session.pool.enable</td>
</tr>
<tr>
<td>dfc.session.pool.timeout</td>
<td>dfc.session.pool.expiration_interval</td>
</tr>
<tr>
<td>dfc.session.surrogate.check.interval</td>
<td>dfc.session.surrogate.check_interval</td>
</tr>
<tr>
<td>dfc.session.surrogate.mode</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.storagepolicy.diagnostics.enabled</td>
<td>dfc.storagepolicy.enable</td>
</tr>
<tr>
<td>dfc.storagepolicy.ignore.rule.errors</td>
<td>dfc.storagepolicy.ignore_rule_errors</td>
</tr>
<tr>
<td>dfc.storagepolicy.validation.interval</td>
<td>dfc.storagepolicy.validation_interval</td>
</tr>
<tr>
<td>dfc.strictURI</td>
<td>dfc.xml.use_strict_uri</td>
</tr>
<tr>
<td>dfc.tracing.baseTraceFileName</td>
<td>dfc.tracing.file_prefix</td>
</tr>
<tr>
<td>dfc.tracing.enabled, r_trace_level</td>
<td>dfc.tracing.enable</td>
</tr>
<tr>
<td>dfc.tracing.entrancePointExprs</td>
<td>dfc.tracing.method_name_filter</td>
</tr>
<tr>
<td>dfc.tracing.loggingMode</td>
<td>dfc.tracing.file_creation_mode</td>
</tr>
<tr>
<td>dfc.tracing.maxFileSize</td>
<td>dfc.tracing.max_file_size</td>
</tr>
<tr>
<td>dfc.tracing.mode</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.tracing.stackDepth</td>
<td>dfc.tracing.max_stack_depth</td>
</tr>
<tr>
<td>dfc.tracing.threadNameExprs</td>
<td>dfc.tracing.thread_name_filter</td>
</tr>
<tr>
<td>dfc.tracing.timestampDateFormat</td>
<td>dfc.tracing.date_format</td>
</tr>
<tr>
<td>dfc.tracing.traceFileDirectory</td>
<td>dfc.tracing.dir</td>
</tr>
<tr>
<td>dfc.tracing.userNameExprs</td>
<td>dfc.tracing.user_name_filter</td>
</tr>
<tr>
<td>dfc.user.dir</td>
<td>dfc.data.user_dir</td>
</tr>
<tr>
<td>dfc.validation.expr.currency.check</td>
<td>dfc.validation.expr.currency_check_interval</td>
</tr>
<tr>
<td>dfc.validation.expr.debug.all</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.validation.expr.debug.code</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.validation.expr.debug.eval</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.validation.expr.debug.tree</td>
<td>No change</td>
</tr>
<tr>
<td>dfc.validation.expr.disable_java</td>
<td>No change</td>
</tr>
</tbody>
</table>

131
### Old name | New name
---|---
dfc.validationoverrides.currency.check | dfc.validation.overrides.currency_check
_interval
dfc.session.recycle_interval, connect_recycle
_interval | dfc.session.reuse_limit
None | dfc.validation.allow.empty_string.list_complete
None | dfc.search.xquery.option.parallel_execution.enable
None | dfc.session.load_balance_strategy
None | dfc.session.max_server_choice_age
None | dfc.session.keepalive.enable
None | dfc.connection.unused_connection_timeout
None | dfc.xml.suppress_default_namespace_decl
None | dfc.internal.purge_far_connections

**Note:** Compatibility is ensured with previous properties that refer to "ecis".

### dmcl.ini key migration to dfc.properties

Table 26, page 132 describes the dmcl.ini keys that migrated to the dfc.properties file since Documentum 6.5.

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

<table>
<thead>
<tr>
<th>dmcl.ini key</th>
<th>Corresponding new dfc.properties key</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_code</td>
<td>dfc.application_code</td>
</tr>
<tr>
<td>batch_hint_size</td>
<td>dfc.batch_hint_size</td>
</tr>
<tr>
<td>backup_host</td>
<td>dfc.docbroker.host</td>
</tr>
<tr>
<td>backup_port</td>
<td>dfc.docbroker.port</td>
</tr>
<tr>
<td>backup_protocol</td>
<td>dfc.docbroker.protocol</td>
</tr>
<tr>
<td>backup_service</td>
<td>dfc.docbroker.service</td>
</tr>
<tr>
<td>backup_timeout</td>
<td>dfc.docbroker.timeout</td>
</tr>
<tr>
<td>castore_write_max_attempts</td>
<td>dfc.content.castore.max_write_attempts</td>
</tr>
<tr>
<td>castore_write_sleep_interval</td>
<td>dfc.content.castore.write_sleep_interval</td>
</tr>
<tr>
<td>client_date_format</td>
<td>dfc.date_format</td>
</tr>
<tr>
<td>client_locale</td>
<td>dfc.locale</td>
</tr>
<tr>
<td>connect_pooling_enabled</td>
<td>dfc.session.pool.enable</td>
</tr>
<tr>
<td>connect_retry_limit</td>
<td>dfc.session.max_connect_retries</td>
</tr>
<tr>
<td>ini_file_path</td>
<td>dfc.config.file</td>
</tr>
</tbody>
</table>
**Table 27**, page 134 lists the *dmcl.ini* keys that are obsolete since Documentum 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 Documentum 6.5 and later versions.

---

<table>
<thead>
<tr>
<th><em>dmcl.ini</em> key</th>
<th>Corresponding new <em>dfc.properties</em> key</th>
</tr>
</thead>
<tbody>
<tr>
<td>local_clean_on_init</td>
<td>dfc.data.local_clean_on_init</td>
</tr>
<tr>
<td>local_diskfull_limit</td>
<td>dfc.data.local_diskfull_limit</td>
</tr>
<tr>
<td>local_path</td>
<td>dfc.data.local_dir</td>
</tr>
<tr>
<td>local_purge_on_diskfull</td>
<td>dfc.data.local_purge_on_diskfull</td>
</tr>
<tr>
<td>max_session_count</td>
<td>dfc.session.max_count*</td>
</tr>
<tr>
<td>primary_host</td>
<td>dfc.docbroker.host</td>
</tr>
<tr>
<td>primary_port</td>
<td>dfc.docbroker.port</td>
</tr>
<tr>
<td>primary_protocol</td>
<td>dfc.docbroker.protocol</td>
</tr>
<tr>
<td>primary_service</td>
<td>dfc.docbroker.service</td>
</tr>
<tr>
<td>primary_timeout</td>
<td>dfc.docbroker.timeout</td>
</tr>
<tr>
<td>ref_binding_label</td>
<td>dfc.reference.binding_label</td>
</tr>
<tr>
<td>secure_connect_default</td>
<td>dfc.session.secure_connect_default</td>
</tr>
<tr>
<td>token_storage_path</td>
<td>dfc.tokenstorage.dir</td>
</tr>
<tr>
<td>token_storage_enabled</td>
<td>dfc.tokenstorage.enable</td>
</tr>
<tr>
<td>umask</td>
<td>dfc.data.umask</td>
</tr>
<tr>
<td>use_compression</td>
<td>dfc.content.use_compression</td>
</tr>
<tr>
<td>use_content_server</td>
<td>dfc.content.use_content_server</td>
</tr>
</tbody>
</table>

*The settings from *dmcl.ini* from your current configuration are transferred after an upgrade. This might cause an issue if the *dmcl.ini* property *dfc.max_session_count* is set to a low session count. EMC recommends that you remove this property setting from the *dmcl.ini* file before performing an upgrade.*
### Table 27. Obsolete session configuration options

<table>
<thead>
<tr>
<th>Entry</th>
<th>Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_during_rpc</td>
<td>dmcl.ini</td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>client_codepage</td>
<td>dmcl.ini</td>
<td>none</td>
</tr>
<tr>
<td>client_os_codepage</td>
<td>dmcl.ini</td>
<td>none</td>
</tr>
<tr>
<td>connect_callback_enabled</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>connect_failure_callback</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>connect_failure_data</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>connect_success_callback</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>connect_success_data</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>content_callback_data</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>content_callback_function</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>local_diskfull_warn</td>
<td>dmcl.ini</td>
<td>none</td>
</tr>
<tr>
<td>network_callback_data</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>network_callback_function</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>new_connection_callback</td>
<td>api config and session config objects</td>
<td>none</td>
</tr>
<tr>
<td>new_connection_data</td>
<td></td>
<td>none</td>
</tr>
<tr>
<td>nfs_enabled</td>
<td>dmcl.ini</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
<tr>
<td>r_trace_file</td>
<td>dmcl.ini</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
<tr>
<td>r_trace_level</td>
<td>dmcl.ini</td>
<td>Implementation now allows per-session caches to dynamically adapt to free memory.</td>
</tr>
<tr>
<td>client_cache_size</td>
<td>dmcl.ini</td>
<td></td>
</tr>
</tbody>
</table>
# dfc.properties Keys

<table>
<thead>
<tr>
<th>Entry</th>
<th>Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect_timeout</td>
<td>dmcl.ini</td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>connect_recycle_interval</td>
<td>dmcl.ini</td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>exception_count</td>
<td></td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>exception_count_interval</td>
<td></td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>terminate_on_exception</td>
<td></td>
<td>Is specific to native code DMCL.</td>
</tr>
<tr>
<td>i_override_list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache_queries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_connection_per_session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>use_local_always</td>
<td>dmcl.ini</td>
<td>Option to use server common area is not available in DFC 6.5, so this becomes unneeded.</td>
</tr>
<tr>
<td>use_local_on_copy</td>
<td>dmcl.ini</td>
<td>Option to use server common area is not available in DFC 6.5, so this becomes unneeded.</td>
</tr>
</tbody>
</table>

## Obsolete dfc.properties Keys

Table 28, page 135 lists the dfc.properties keys that are obsolete since version 6.5. Setting these keys has no effect on DFC 7.x.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfc.tracing.combineDMCL</td>
<td>dfc.properties</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
<tr>
<td>dfc.tracing.compactMode</td>
<td>dfc.properties</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
<tr>
<td>Entry</td>
<td>Source</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dfc.tracing.recordParameters</td>
<td>dfc.properties</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
<tr>
<td>dfc.tracing.recordReturnValue</td>
<td>dfc.properties</td>
<td>Replaced by new tracing implementation—refer to the <em>EMC Documentum Administrator User Guide</em> for information.</td>
</tr>
</tbody>
</table>
Appendix D

LDAP Certificates

When you upgrade to 7.2 from a previous version in Microsoft Windows, you may need to import the LDAP certificates again as the hash value might have changed. Use DA to import the LDAP certificate again. If you need to import this certificate manually, then perform the following steps:

**Note:** The name of the LDAP certificate depends on the hash value generated for it.

1. `openssl x509 -hash -in <filename>`
   The openssl version that is recommended is 1.0.0 or higher.

2. Attach Process Monitor and add the following values:
   - `filter process name = "documentum.exe"
   - `event class = "file system"
   - `exclude "<docbase>.log"

   Run the LDAP synchronization job and try authenticating an LDAP server user that has got synced. Search for `ldapdb` and fetch the hashvalue that is required. Rename the certificate file with this hash value.

3. Modify the `dm_start_<repo>` file by adding the prefix `truss` (Solaris) or `strace` (AIX/Linux).
   Run the LDAP synchronization job and try authenticating an LDAP server user that got synced. Search for `ldapdb` and fetch the hashvalue that is required. Rename the certificate file with this hash value.
6.7 clients
   dm_bof_registry, 94
64-bit Content Server
   upgrading to, 37

A
ACS, 29
aek.key file, 28

B
BOCS, 29
BOCS servers
   described, 47
BOF, 94
BOF2 modules
   migrating, 91

C
CLEAN_LINKS, 97
CMIS
   customizations, 103
   getFolderParent, 103
configuration
   departmental system, 26
   enterprise system, 26
consistency checker utility, 79

D
database
   changing location, 35, 38
   changing version, 35, 38
   upgrading DB2 database, 36
database_refresh_interval key, 36
databases
   requirements for upgrade, 36
   UTF-8, 36
DFC
   deployment, 96
   methods, 112
   migrating, 93
   search, 95
DFC classes
   migration, 93
dfc.properties, 129
   DfPreferences, 96
dmcl.ini key migration, 132
   key names, 129
   obsolete dmcl.ini keys, 133
   obsolete keys, 135
DfPreferences
   JMX management, 96
DFS
   migrating, 99
differences
   DFC Java classes, 93
distributed configurations
   upgrading, 45
distributed content, 30
dm_bof_registry, 27
dm_extern_file, 97
dm_linkedstore, 97
DMCL APIs, 111
   migrating, 95
dmi_linkrecord, 97
DocApps
   migrating, 91
DQL
   migrating content, 91
DSN, 38
dump and load, 35, 38

F
files
   aek.key, 28

G
global registry
Index

defined, 27

I
installation order
new system, 15
internationalization
UTF-8, 36
IPv4 configuration, 94

L
linked store storage areas, 97

M
migrating
BOF2 and DocApps, 91
migrating Content Server, 77
6.6 to 7.1, 105
checklist, 79
custom methods, 91
migration methods, 81
migration process, 77
migration utility, 90
planning, 80
post-migration tasks, 92
scenarios, 105
Windows/SQL Server platform, 105
XML store, 90
migration
client first, 29
overview, 14
modules
migrating, 91

O
object replication, 30
object types
changed, 120
new, 119
new properties, 127
ODBC, 38
optical storage devices, 98

P
Planning worksheet
application server host, 24
client machine, 25
Content Server host, 23
customized components, 25
index server host, 24

R
repositories
copying, 28
federation, 30
precopying tasks, 28
repository copies, 28
repository federations
upgrading, 30
root password, 35

S
SBO, 94
SQL Server 2005, 85
system updates
order, 16

T
TBO, 94
Terminal Services, 45
testing an upgrade, 28
third-party software, 27

U
upgrade
overview, 13
strategies, 19
Upgrade
BOCS servers, 47
upgrading Content Server
6.5 SP3 to 7.1, 51, 59
6.7 SP1 to 7.1, 64
6.7 SP2 to 7.1, 68
6.7 to 7.1, 63
7.0 to 7.1, 72, 75
active automatic tasks, 36
checklist, 33
cluster services, 44
custom plug-ins, 40
database_refresh_interval key, 36
described, 33
distributed configurations, 45
distributed content, 30
dm_server_config object, 37
effects on system, 33
errors, 39
FAST, 39
Linux/Oracle platform, 64, 75
object replication, 30
post-upgrade, 48
Red Hat Linux 5.x, 40
repository federations, 30
scenarios, 51
software, 40
Solaris/Oracle platform, 68
testing, 28
UTF-8, 36
Windows/Oracle platform, 72
Windows/SQL Server platform, 51, 59, 63

W
Windows requirements
Terminal Services, 45