

EMC[®] Documentum[®] Platform and Platform Extensions

Version 7.3

Installation Guide

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

Legal Notice

Copyright © 1994–2017 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	31
Chapter 1 Content Intelligence Services	33
Introduction	33
Components	33
Compatibility	34
Preinstallation tasks	34
Installation checklist	35
Prepare the required information	35
Additional preinstallation tasks for Linux hosts	36
Additional preinstallation tasks for Windows hosts	37
Migrating status information for document sets	37
Installing Content Intelligence Services	38
Installing CIS in silent mode	41
Installing multiple CIS instances	42
Setting up Docker for CIS	42
Installing and configuring CIS on Ubuntu Docker container	42
Configuring CIS on Docker container	43
Installing CIS on Docker container	44
Common Notes for Docker environment	45
Sample of silent.ini for installing CIS	45
Completing the Installation	46
Authenticated connection between CIS and the repository	46
Deploying CIS artifacts (DAR file) manually	47
Enabling the repository for CIS	47
Validating the Installation	49
Verifying the deployment of CIS artifacts (DAR file)	49
Verifying that the repository is enabled for CIS	49
Verifying that the tables are created	50
Verifying the configuration of the entity detection server	50
Verifying that all services are started	50
Troubleshooting Installation Issues	51
Modifying the ports for the entity detection server	51
Some Luxid services (4/7) are not started	52
Problem	52
Cause	52
Resolution	53
CIS installer unable to connect to Content Server	53
Problem	53
Resolution	53
Uninstalling Content Intelligence Services	53
Uninstalling (Windows hosts)	53
Uninstalling (Linux hosts)	54
Downgrading Content Intelligence Services	54

Chapter 2	Content Management Interoperability Services	55
	Introduction	55
	Configuration settings	56
	General JVM Configuration Settings	56
	Using urandom Generators on Linux Systems	56
	Documentum CMIS Configuration Files	57
	DFC Configuration	57
	Documentum CMIS Runtime Properties	58
	Anonymous Access Settings	62
	Maximum Items Default and Upper Limit Settings	63
	Configuring Kerberos SSO	63
	Enabling Kerberos SSO	64
	Configuring the Documentum CMIS Web Application's SPN and *.keytab File	64
	Mapping the SPN to a User Name	65
	Configuring the Application Server for Kerberos	66
	Configuring krb5.ini and cmis-runtime.properties Files	66
	Configuring the JAAS.conf file	67
	Configuring the Documentum CMIS Web Application	70
	Logging for Kerberos	72
	Performance Best Practices	72
	QUEST TCP/UDP Settings	72
	Configuring token-based authentication for Browser Binding	73
	HttpBasicAuthentication with Tokens	73
	Kerberos SSO with Tokens	74
	Deploying to supported application servers	75
	Apache Tomcat	75
	VMware vFabric tc Server	75
	Oracle WebLogic Server	75
	IBM WebSphere	76
	Post deployment	77
	Validation	77
	RESTful AtomPub Service Document	77
	Web Service Entry Points	77
	Browser Binding URLs	78
Chapter 3	Content Server	79
	Installation overview	79
	Content Server architecture	79
	Content Server and repository	79
	Connection broker	80
	Global registry	81
	Global registry user	81
	Content Server installation models	82
	Basic installation model	82
	Planning the installation	83
	Location of the content storage area (data directory)	86
	Ports to reserve for Content Server	87
	Connection modes to connect to connection broker and repository	87
	Content Server optional modules	87
	Planning the system size	89
	Pre-installation requirements and tasks	89
	System requirements	89
	General database requirements	89
	Requirements for Microsoft SQL Server	91
	Requirements for Oracle database	91

Requirements for DB2 database	92
Requirements for DB2 performance wizard	93
Requirements for PostgreSQL database	93
Enabling data partition in PostgreSQL database	95
Tuning PostgreSQL database	97
Setting up required user accounts	98
Installation owner account	98
Installation owner account naming requirements	99
Required rights for the installation owner account	99
Installation owner's email account and SMTP server information	100
Repository owner account	100
Repository user accounts	101
Pre-installation tasks on Windows	102
Pre-installation tasks on UNIX and Linux	102
Performing pre-installation tasks for SUSE Enterprise Linux	102
Performing pre-installation tasks for Red Hat Enterprise Linux 7.x	102
Setting the required environment variables	103
Setting up the services file	104
Installing the JCE policy files	105
Enabling random generator on Linux	105
Preparing relational database management system (RDBMS)	106
Creating an ODBC data source for Microsoft SQL Server	106
Configuring Oracle database	106
Configuring the tnsnames.ora file	106
Performing additional Oracle database configuration tasks	107
Configuring DB2 database	107
General guidelines	107
Configuring DB2 from Control Center	107
Configuring DB2 database from command line	108
Tuning DB2 database	109
Running multiple Content Servers on DB2 host	110
DB2 repository sizes	110
Configuring internationalization settings	111
Installing MailApp DAR	111
Preparing for remote key management	111
DPM overview	112
DPM limitations	112
Identities, identity groups, and key classes	113
Identities	113
Identity groups	114
Key classes	115
Acquiring certificates	115
Preparing DPM for remote key management	115
Information required to configure repository	118
Preparing installation package	118
Configuring for Certificate-based SSL communication	118
Connection modes	119
Prerequisites	119
Certificates	120
Building a Keystore	120
Preparing Keystore password	120
Building a Trust Store	121
Location of the keystore file, keystore password file, and trust store file	121
Configuration	121
Configuring connection broker	121
Configuring Server	122
Configuring DFC	123

Installer support.....	123
Compatibility	123
Constraints and limitations	124
Installing and configuring Content Server	124
Installation workflow	124
Installing and configuring Content Server	124
Using GUI.....	124
Installing Content Server program files	125
Creating a connection broker.....	126
Creating a repository	127
Viewing the configuration summary	132
Viewing the version details of installed products or components	132
Using command line.....	133
Creating the silent installation files	133
Running the installation and configuration from the command line	134
Uninstalling Content Server from the command line.....	135
Supporting Docker for Content Server	136
Introduction.....	136
Supported Docker configurations	136
Installing Docker	136
Common notes	137
Creating the Content Server Linux/Oracle Docker image.....	137
Prerequisites	137
Hardware requirements (Machine 1 – Container for Content Server).....	138
Hardware requirements (Machine 2 – Container for Oracle server)	138
Software requirements	138
Configuring the Red Hat Enterprise Linux base image	138
Installing Oracle client	139
Configuring to create the Content Server Linux/Oracle Docker image	140
Exporting environment variables for storing passwords to Docker environment	141
Installing and configuring Content Server on Docker environment	141
Upgrading Content Server using seamless upgrade method	142
Installing and configuring of Content Server HA on Docker environment	142
Completing the installation	143
Reviewing Content Server installation and configuration logs	143
Configuring symbolic link path for Java	143
Enabling the purge audit job	144
Post-installation tasks	144
Configuring WildFly for SSL	144
Supported communication configurations.....	145
Changing the connection mode from native to secure	146
Backing up keystore files.....	147
Changing the default passphrase.....	147
Binding Content Server to a network card	147
Installing Content Server client.....	148
Installing Content Server using an existing database account.....	148
Enabling xPlore search for new emails	148
Using the migration utility	149
Changing the repository ID.....	150
Changing the repository name.....	150
Changing the server configuration name.....	151

Changing the hostname of the Content Server machine.....	151
Changing the installation owner	151
Running the migration utility	152
Getting Started with Content Server.....	152
Starting/stopping Content Server on Windows	152
Starting Content Server on UNIX and Linux.....	153
Java Method Server for High-Availability	154
Overview	154
Installation of additional JMS	154
Enabling JMS HA feature	155
JMS HA configurations	155
Method location options for JMS HA	156
Prerequisites	156
JMS status	157
Enabling and understanding logs	157
Error messages.....	158
Supported HA configurations.....	158
Content Server and two or more Java Method Servers on a single host	159
JMS for HA on multiple hosts.....	160
Two or more Content Servers and two or more Java Method Servers on a multiples host	161
Installing Content Server with Microsoft Cluster Services.....	161
Microsoft Cluster Services overview	161
Choosing a configuration	162
Pre-installation requirements	164
Configuring an active/passive cluster.....	164
Creating the cluster resource group	164
Installing Content Server software on the nodes	165
Configuring Content Server.....	165
Configuring the connection brokers	166
Creating additional cluster resources on Microsoft Cluster Services	167
Verifying failover.....	168
Configuring an active/active cluster	168
Creating the first cluster resource group.....	168
Installing Content Server software on the hosts.....	169
Configuring Content Server on the first and second nodes.....	169
Configuring the second cluster resource group	169
Modifying server.ini and dfc.properties	169
Configuring the connection broker and repository for failover.....	169
Verifying failover.....	170
Uninstalling Content Server	170
Uninstalling components	170
Deleting a repository	171
Deleting a connection broker.....	171
Uninstalling the Content Server software	171
Troubleshooting	171
Identifying the problem and resolution	172
While installing on Linux, Installer hangs when the number of mount points exceeds 4000	175
Recovering from a failed repository configuration or upgrade	176
Recovering from a stalled Content Server upgrade.....	176
Identifying issues for Certificate-based SSL communication	177
Connection broker startup fails.....	178
Server startup fails.....	178
Server not able to connect to connection broker	179
Clients are unable to connect to the connection broker	179

Clients are unable to connect to the Server.....	180
Additional information	180
Content Server error messages and causes.....	180
Recovering from a failed filestore configuration	186
Content Server installation directories and repository configuration scripts.....	186
Content Server installation file structure.....	187
uninstall.....	187
data	187
dba	188
fulltext.....	188
product.....	188
server_uninstall.....	188
share.....	188
Additional directories	189
Scripts run during installation or upgrade	192
Configuration objects	195
Object type categories for Oracle database storage	196
Type categories for tablespace specifications.....	196
Type categories for extent allocation	197
Object types categorized as large	197
Object types categorized as small.....	197
Object types categorized as default	198
Defining Oracle or DB2 database parameters for repository tables	199
Defining the tablespace	199
FUNCTION_SPECIFIC_STORAGE	200
TYPE_SPECIFIC_STORAGE	200
Defining the Oracle extent sizes.....	201
Changing storage parameters for individual object types on Oracle	202
Changing storage parameters for categories of types on Oracle.....	202
User-defined object types	203
Chapter 4 Distributed Content	205
Distributed Configuration components	205
Building blocks.....	205
Network locations	205
Benefits and best use.....	206
Configuration requirements	206
ACS servers	206
Benefits and best use.....	206
Configuration requirements	207
ACS caching.....	207
BOCS servers	208
Benefits and best use.....	208
Limitations.....	208
BOCS encryption	209
Partial download of content	209
Configuration requirements	209
DMS servers.....	210
Benefits and best use.....	210
Pre-cached content	210
Benefits and best use.....	210
Limitations.....	210
Configuration requirements	210
Asynchronous write capabilities	211
Benefits and best use.....	211
Limitations.....	212

Configuration requirements	212
Distributed storage areas and configuration requirements.....	212
Benefits and best use.....	213
Limitations.....	213
Configuration requirements	213
Remote Content Servers	213
Benefits and best use	214
Limitations.....	214
Configuration requirements	214
Shared content	214
Benefits and best use.....	215
Configuration requirements	215
Content replication.....	215
Benefits and best use	216
Configuration requirements	216
Reference links	216
Benefits and best use.....	216
Configuration requirements	217
Object replication	217
Benefits and best use	218
Configuration requirements	218
Federations	219
Benefits and best use.....	219
Configuration requirements	219
Distributed Configuration models	220
Overview of models.....	221
Single-repository distributed models	221
Single model 1: Single repository with content persistently stored at primary site and accessed using ACS or BOCS servers	221
Benefits and best use.....	224
Single model 2: Single repository with content in a distributed storage area.....	224
Benefits and best use.....	226
Building block architectures for single-repository models.....	226
Network locations	226
ACS servers	226
BOCS server.....	227
Push and pull modes	228
Repository inclusion or exclusion	228
Asynchronous write configuration.....	229
DMS servers.....	229
Content pre-caching	229
Asynchronous write	229
Communication flow descriptions	230
Communication flow when a remote user requests a document for viewing.....	230
Communication flow when a remote user updates a content file	231
Asynchronous write with BOCS server	231
Synchronous write with BOCS server.....	232
Synchronous write with ACS server.....	232
Communication flow when a content pre-caching request occurs	233
Implementation of distributed storage areas.....	233
Proximity values.....	234
Use by Content Servers	234
Use by ACS servers.....	235
Use by network locations	235
Setting up an ACS server for load balancing and failover	235

Shared content	236
Content replication	236
Building block architectures for multirepository models	237
Reference links	237
Object type implementation	237
Mirror objects	237
Replica objects	238
Reference objects	238
Valid object types for reference links	238
Reference link binding	238
Reference link storage	239
Type-specific behavior of reference links	239
Reference link updates	239
Operations on replica objects	239
Reference link security	240
Mirror objects	240
Replicas	241
Object replication	241
Replication jobs	241
Multi-dump file replication jobs	241
Best use of multiple dump file replication	242
Conditions of use	242
Replication modes	242
Nonfederated replication mode	242
Federated replication mode	243
What is replicated	244
Aspect modules and replication	246
Format objects and replication	246
Data dictionary information and replication	246
Display configuration information and replication	246
Full refreshes	247
Related objects storage	247
How the replication process works	247
Federations	247
Supporting architecture	248
Jobs and methods	248
Multi-repository distributed models	249
Multiple repositories using object replication	249
Multiple repositories working as a federation	249
Distributed environments and the secure connection defaults	250
Distributed messaging	251
Installing Documentum Messaging Services (DMS) servers	252
Introduction	252
Preinstallation requirements	252
Installing a DMS server	253
Configuring a DMS server	257
Timing the pushing of messages to BOCS servers	258
Best practices	258
Database settings for the Oracle-based DMS database	258
DMS configuration	259
BOCS configuration	260
Starting and stopping a DMS server	260
Uninstalling a DMS server	260
Installing BOCS	261
Overview	261
BOCS server environment	261
Preinstallation requirements	261
Installing BOCS	262

Configuring BOCS	263
Configuration requirements	263
DMS and global registry compatibility requirements	264
The acs.properties file	264
acs.properties file location	264
File administration	264
Changing the JMX user password	265
Default settings	265
Configuration keys that cannot be changed	265
Configuration keys that to be modified only as a part of BOCS reconfiguration.....	266
Configuration keys that you can change.....	266
Adding or modifying a root cache directory	266
Defining the parked content directory	267
Configuring cache housekeeping	267
Configuring consistency checks	268
Configuring cache write intervals	268
Configuring use of content retrieval URLs	269
Log files	269
Application server log file	269
BOCS-specific DFC log file	269
Log file for all DFC messages	270
Log file for recording BOCS messages	270
Log file size and backups	270
Reconfiguring a push BOCS server to pull mode.....	271
Enabling BOCS access from behind a firewall	272
Enabling or disabling the write mode.....	272
DNS requirement for web-based client hosts in distributed environment	273
Creating a BOCS configuration object	273
Configuring security for BOCS servers in pull mode	273
Installing and configuring BOCS on Docker environment.....	273
Upgrading BOCS	273
Removing BOCS.....	274
On Windows	274
On UNIX and Linux	274
Starting and stopping BOCS.....	274
On Windows	274
On UNIX and Linux	274
Configuring WildFly, ACS, BOCS, and DMS for Secure Socket Layer (SSL) connections	275
Generating and importing security certificates.....	275
Configuring web applications for SSL	277
ACS.....	278
BOCS.....	278
DMS	279
WDK Client Application Server.....	279
Modify/configure ACS/BOCS/DMS configs	280
Troubleshooting the SSL Configuration.....	280
Installing remote Content Servers in distributed or load-balanced configurations	281
Preinstallation requirements.....	281
Installing and configuring the remote Content Server.....	282
Upgrading a distributed or load-balanced configuration	283
Deleting a remote Content Server	284
Implementing single-repository models	284
Implementing a distributed repository without a distributed storage area.....	284

Installing with distributed storage areas.....	285
Planning	285
Guidelines	286
Estimating disk space	286
Estimating document size	287
An example of disk space calculations	287
Setting up the sites.....	287
The dm_rcs_setup.ebs script.....	290
Creating network locations.....	290
Adding network locations to an ACS or BOCS configuration object	291
Projecting an ACS server to connection brokers	291
Setting ACS proximity values for network locations.....	292
Defining accessible storage areas for an ACS server	293
Accessing file stores in a distributed environment	294
Modifying an acs.properties file.....	294
The mode.cachestoreonly entry	295
Adding entries for additional servers	295
Disabling access to an ACS server	295
Configuring shared content files	296
Creating pre-caching content jobs	296
Setting up content replication	298
Deciding which tool to use	298
Automatic Replication	298
Manual replication	298
Using the surrogate get feature.....	299
The dm_SurrogateGet method.....	299
Using REPLICATE.....	300
Using IMPORT_REPLICA.....	300
Setting proximity values for Content Server projection to a connection broker.....	301
Guidelines	301
Example of selecting proximity values	301
At site A	302
At site B.....	302
At site C.....	303
First-time use	303
Implementing multirepository models.....	303
Repository configuration for distributed environment	303
Connection broker setup	303
User setup.....	304
Password setup	305
Object replication jobs.....	306
Distributed operations job activation.....	306
Setting up a federation	306
Choosing the governing repository	306
Identifying user subtypes for propagation.....	307
Creating a federation	307
Implementing object replication	308
Defining business requirements.....	308
Functional divisions and groups	309
Document types	309
User distribution and geography	310
Security	310
Infrastructure	311
Reference metrics	311
Network replication options	312
Replication system administration	312
Determining computing resources	313

Determining needed jobs	313
Disk space requirements	314
For replicated documents	314
Temporary space for dump files	315
Job scheduling	316
Handling overlapping jobs	317
Site setup	317
Connection broker setup and validation	317
Macintosh access protocol	318
Disk space for temporary files	318
Content storage	318
Cabinets and folders	318
Defining jobs	319
Guidelines for all jobs	320
Guidelines for multidump file jobs	320
Setting up tracing	320
Manual dump file transfers	320
Best practices for object replication	321
Managing single-repository models	321
Adding a distributed component	321
Removing a distributed component	323
Removing files from component storage areas	323
Using DQL EXECUTE	323
Troubleshooting surrogate get	323
Tracing surrogate get	323
The trace file	324
Turning on trace file generation	324
Tracing method invocations	324
Resolving problems	324
The only_fetch_close property	324
The get_method properties	325
Connection broker projection targets	325
Time settings	325
Overriding remote content server use	325
Using the use_content_server key	326
Using the connection request	326
Using both use_content_server key and the connection request	326
Login failures in remote content server setups	329
Tuning query performance	329
Managing multirepository models	329
Manipulating a federation	329
Adding a member	329
Removing a member	330
Destroying a federation	330
Inactivating a governing repository	330
User operations	330
Creating a user	331
Modifying user information	332
Renaming a user	333
Making a local user global	333
Using Documentum Administrator	333
Using DQL	333
Making a global user local	334
Deleting a global user	334
Group operations	334
Creating a group	334
Modifying a group	335
Renaming a group	335

Deleting a group.....	335
Making a global group local	335
Modifying object replication jobs	336
Obtaining a list of jobs	336
Scheduling federation jobs	336
Identifying the federation jobs operator	337
Tracing ACL replication in federation jobs.....	337
Job reports and log files.....	337
Job reports	337
Job log files	338
The dm_DistOperations job.....	338
Monitoring and debugging federation jobs.....	340
Recovering from replication job failures	340
Clearing the replicate_temp_store storage area	341
Handling replicas	341
Defining a binding label for reference links	341
Determining whether an object is a replica	341
Determining a replica's source	342
Federation infrastructure	342
Chapter 5 Documentum Administrator	345
Planning for deployment	345
Required and optional supporting software.....	345
Typical configuration	345
Application server host requirements.....	346
Customizing Documentum Administrator	346
Preparing the client hosts	347
Ensuring a certified JVM on browser clients	347
Enabling HTTP content transfer in Internet Explorer.....	347
Preparing the application server host	347
Application servers.....	347
Setting the Java memory allocation	348
Turning off failover.....	348
Preparing environment variables for non-default DFC locations	349
Configuring Apache Tomcat.....	349
Configuring JBoss EAP	350
Configuring VMware vFabric tc Server	351
Preparing IBM WebSphere	351
Disabling HttpOnly Property	351
Supporting failover in a cluster.....	351
Applying policies for IBM WebSphere security	351
Preparing Oracle WebLogic.....	353
Disabling HttpOnly property	353
Preparing the application server for Java 2 security.....	353
Preparing to use an external web server	354
Deploying Documentum Administrator.....	354
Prerequisites	354
Deploying the WAR file	355
Enabling DFC memory optimization.....	359
Configuring UCF	359
Forcing UCF to install a configured JRE	359
Enabling retention of folder structure and objects on export	360
Enabling external searches	360
Configuring the connection to the search server.....	360
Configuring the connection to the backup search server	360
Requirement for full-text indexing.....	361
Resource Management availability.....	361

Enable presets for Administrator Access and Resource Management	361
Modal popup	361
Configuring the modal popup	362
Deploying and configuring Documentum Administrator on Docker environment	362
Post-deployment tasks	363
Configuring IBM WebSphere.....	363
Configuring Oracle WebLogic class loading behavior.....	363
Configuring UCF on Oracle WebLogic Server.....	364
Configuring single sign-on for security servers	364
Configuring IBM WebSEAL single sign-on (SSO) authentication.....	366
Prerequisites	367
Configurations in custom/app.xml file to enable IBM WebSEAL authentication	367
Configuring Kerberos authentication	367
Kerberos-based single sign-on authentication in Documentum Administrator	368
Prerequisites	368
Configurations in custom/app.xml file to enable Kerberos authentication	368
Enabling Kerberos SSO authentication in Documentum Administrator	368
Configuring the Kerberos domain name.....	369
Configuring Kerberos fallback.....	369
Sample Kerberos configuration in app.xml	369
Preparing Documentum Administrator and the browser to meet Kerberos SSO setup requirements	370
Create user account for Documentum Administrator in the active directory.....	370
Define a Service Principal Name for Documentum Administrator and create KeyTab file.....	370
Configuring the client browser to use the SPNEGO protocol	370
Creating JAAS configuration file.....	371
Creating a configuration file for the application server to connect to the KDC server	373
Application Server-specific configurations.....	374
Apache Tomcat.....	374
Oracle WebLogic	374
IBM WebSphere.....	374
Cross-frame scripting configuration.....	375
Starting Documentum Administrator.....	375
Maintenance and procedures.....	375
Logs to monitor	376
Application Server	376
Content Server repository	376
Java Method Server.....	376
Index Server.....	376
Disk space management.....	377
Jobs	377
DQL queries.....	377
Network connectivity interruption.....	378
RAM and CPU Utilization maxed out	378
Sessions to monitor.....	378
Security and Server access maintenance	378
Improving Performance	379
Java EE Memory Allocation.....	379
Preferences.....	380
Browser History	380

	Value Assistance.....	381
	Search Query Performance	381
	High Latency and Low Bandwidth Connections	381
	Qualifiers and Performance	383
	Import Performance.....	383
	Load Balancing.....	383
	Modal Windows and Performance.....	384
	Troubleshooting deployment.....	384
	Wrong JRE used for application server	384
	No global registry or connection broker	384
	No connection to repository	385
	Login page incorrectly displayed	385
	Slow performance.....	385
	Out of memory errors in console or log	385
	Slow display first time	385
	DFC using the wrong directories on the application server	386
	Tag pooling problem.....	386
	UCF client problems	386
	Connection issues between a Federated Search server and IPv6 clients	387
	Max Sessions error.....	387
Chapter 6	Documentum Foundation Classes	389
	Before you install DFC	389
	Where to install DFC	390
	Java 2 Security	390
	Removing old files.....	390
	Removing old DFC	391
	Whether to upgrade client programs.....	391
	Establishing the environment for DFC	391
	Defining file system locations for DFC components	392
	DFC program root directory	392
	DFC user root directory	392
	Directory for shared libraries.....	392
	Directory for DFC configuration files	392
	Locations of DFC classes	393
	Setting environment variables	393
	Using the DFC config directory	394
	Uninstalling DFC.....	395
	Uninstalling from Windows	395
	Uninstalling from Linux.....	396
	Installing DFC	396
	Installation requirements	396
	Installing DFC on a Windows system.....	397
	Installing silently	398
	Creating the configuration file	398
	Running the installation program silently	399
	Installing and configuring DFC on Docker environment	399
	IPv6 support	399
	Documentum client connection process (dual-stack mode)	399
	Configuring the client DFC	400
	Configuring the Java Virtual Machine for IPv4 only	400
	Troubleshooting installer problems	401
Chapter 7	Documentum Foundation Services	403
	Introduction.....	403

About DFS deployment	403
Local and remote DFS applications	403
Supported environments.....	403
Web services archive files	404
Clustered deployment for load balancing.....	404
Failover not supported.....	404
Configuration.....	404
General JVM configuration settings	405
MTOM content transfer mode settings	405
DFC configuration	405
Docbroker and global registry properties	406
Properties required by the Schema service.....	406
Reusing privileged DFC client instance	406
Trusted login on the same host as Content Server	407
Configuring dfs-runtime.properties	407
Required settings.....	407
Configuration settings in dfs-runtime.properties	408
Single sign-on properties.....	408
Configuring SSL for DFS.....	409
Deployment on web application servers.....	411
Apache Tomcat.....	411
VMware vFabric tc Server	411
Oracle WebLogic Server	411
IBM WebSphere.....	412
Deploying DFS Java productivity layer on IBM WebSphere 8.5	413
JBoss web application server	414
Validating DFS deployment.....	414
Deploying and configuring DFS on Docker environment	414
Enabling Kerberos authentication	415
Overview	415
Procedure to enable Kerberos SSO	415
Configuring the DFS server's service principal name and *.keytab file	416
Mapping the SPN to a user name.....	416
Configuring the DFS application server for Kerberos	417
Kerberos configuration	418
JAAS configuration.....	418
Enabling Kerberos for DFS remote services	423
Kerberos Diagnostics	426
Enabling SAML authentication in DFS services.....	426
DFS server-side changes.....	426
DFS client-side changes that are using DFS SDK APIs (productivity layer).....	426
Method 1	427
Method 2	427
DFS client-side changes that are not using DFS SDK APIs	428
IBM Tivoli Access Manager for e-business WebSEAL integration.....	431
Implementing Custom SOAP Handler for DFS	431
Configuring DFS Server to use Custom SOAP Handler	432
Preserving JSESSIONID cookie name.....	432
Chapter 8 Federated Search Services	435
Federated Search Services installation	435
Pre-Installation Tasks	435
Preparing Installation Packages	435
Pre-Installation Tasks on UNIX and Linux.....	436
Pre-Installation Tasks on Windows	437

Installing Federated Search Services.....	437
Installation Checklist	437
Running the Installation.....	438
Troubleshooting	439
Upgrading Federated Search Services	439
Uninstalling Federated Search Services	439
On Windows	439
On UNIX and Linux	439
Running the Installation in Silent Mode	440
Starting the Federated Search Services Server	440
On Windows	440
Stopping the Federated Search Services Server	441
Log Files	441
On UNIX and Linux	441
Starting the Admin Center	442
On Windows	442
Stopping the Admin Center.....	442
Log Files	442
On UNIX and Linux	442
Silent Installation Files	443
silent.txt	443
Setting up Docker on Federated Search Services server	444
Installing and configuring FS2 on Ubuntu Docker container	444
Configuring Federated Search Services Server on Docker container	445
Installing FS2 on Docker container.....	445
Common notes for Docker environment.....	445
Federated Search Services adapters installation	446
Overview	446
Main Concepts	447
Adapters.....	447
Adapter Bundles	447
Backends	447
Configuring Adapter Backends	448
Configuring and Testing an Adapter Backend	448
Configuring the Authentication.....	448
Common Backend Configuration Properties.....	449
Common FS2 Attributes.....	460
Documentum Adapter	463
Principles of the Documentum Adapter	464
Installing the Documentum Adapter.....	464
Updating the Documentum Adapter.....	464
Setting the Documentum Adapter	465
Creating a Backend for the Documentum Adapter	465
Configurable Properties	465
Mandatory Properties	465
Optional Properties	466
Documentum eRoom Adapter.....	473
Principles of the eRoom Adapter	474
Installing the eRoom Adapter.....	474
Installing the FS2 Script for eRoom	474
Installing the Web Service-Based Adapter	475
Updating the eRoom Adapter.....	476
Setting the eRoom Adapter	476
Creating an Adapter Backend for eRoom	476
Configurable Properties	477
Mandatory Properties	477
Optional Properties	478
Documentum ApplicationXtender Adapter	482

Principles of the ApplicationXtender Adapter	482
Source Software Requirements	482
Installing the ApplicationXtender Adapter	482
Updating the ApplicationXtender Adapter	483
Setting the ApplicationXtender Adapter	483
Creating an adapter backend for ApplicationXtender	483
Configurable Properties	484
Mandatory Properties	484
Optional Properties	485
Query Translation	489
Full-Text Searches	489
Field Searches	489
Troubleshooting	490
SourceOne Adapter	490
Principles of the SourceOne Adapter	491
Support of SourceOne 6.5	491
Support of SourceOne 6.6, 6.8, and 7.0	491
Installing the SourceOne Adapter	491
Configuring Internet Information Services (IIS) Manager	491
Updating the SourceOne Adapter	492
Setting the SourceOne Adapter	492
Creating an adapter backend for SourceOne	492
Configurable Properties	493
Mandatory Properties	493
Optional Properties	494
Query Translation	497
Constraints Operators	497
Supported Operators	499
Known Limitation	499
Troubleshooting	499
Messages Cannot Be Imported	499
Empty Messages Are Imported	499
Login Name and/or Password Cannot Work When Containing Spaces	500
JDBC/ODBC Adapter	500
Principles of the JDBC/ODBC Adapter	500
Installing and Configuring the Database Drivers	501
Using the JDBC/ODBC Bridge	501
Using a Direct JDBC Driver	501
Updating the JDBC/ODBC Adapter	502
Setting the JDBC/ODBC Adapter	502
Creating an Adapter Backend for JDBC/ODBC	502
Configurable Properties	503
Mandatory Properties	503
Optional Properties	505
OpenSearch Adapter	509
Principles of the OpenSearch Adapter	509
Compatibility	510
Installing the OpenSearch Adapter	510
Updating the OpenSearch Adapter	510
Setting the OpenSearch Adapter	510
Creating an Adapter Backend for OpenSearch	511
Configurable Properties	511
Mandatory Properties	511
Optional properties	512
Troubleshooting	512
Connecting to a Source Using HTTPS	513
Configuring the Adapter with HTTPS	513
Wrapping the Remote Source	513

Checking the Remote Source Certificate	514
Saving the Remote Source Certificate	514
Adding the Certificate to FS2 Keystore File.....	515
xPlore adapter installation.....	516
Introduction.....	516
Installing xPlore Adapter	516
Installation Requirements	516
Configuring xPlore Server.....	516
Installing the Adapter Bundle.....	517
Adapter Properties	517
Creating a Backend.....	517
Configurable Properties	518
Mandatory Properties.....	518
bundle	518
host	518
client	518
protocol	519
port	519
queryLocale	519
domain	519
rootPath.....	519
timeout	520
stopLimit	520
compoundScore	520
query.....	520
result	520
Optional Properties	521
filter.....	521
collection	521
mapout.<FS2 attributes>.....	521
Troubleshooting	522
InfoArchive adapter installation	522
Introduction.....	522
Installing the Adapter.....	522
Installation Requirements	522
Configuring InfoArchive Server.....	522
Installing the Adapter Bundle.....	523
Adapter Properties	523
Creating a Backend.....	523
Configurable Properties	524
Mandatory Properties.....	524
bundle	524
host	524
hasSecurity	524
loginName.....	525
loginPassword.....	525
consumerApplication	525
roles	525
holdingName.....	525
channelName.....	526
locale.....	526
resultSchema.....	526
query.....	526
result	526
mapin	527
mapout.....	527
Optional Properties	527
filter.....	527

compoundScore	527
stopLimit	528
dateFormat	528
Troubleshooting	528
Query Translation.....	528
Google Search Appliance adapter installation.....	529
Introduction.....	529
Installing the Adapter	529
Installation Requirements	529
Installing the Adapter bundle	530
Adapter Properties	530
Mandatory Properties	530
bundle	530
client	530
host	530
output	531
proxystylesheet	531
site	531
Optional Properties	531
client.overview.....	531
image	532
port	532
query.....	532
querySuffix	532
result	532
stopLimit	533
Sample Backends.....	533
Microsoft Exchange adapter installation.....	534
Introduction.....	534
Installing the Adapter	534
Installation Requirements	534
Configuring Microsoft Exchange Server	534
Installing the Adapter Bundle.....	534
Adapter Properties	535
Creating a Backend.....	535
Configurable Properties	535
Mandatory Properties	536
bundle	536
host	536
Optional Properties	536
stopLimit	536
mailboxFolder	536
messageRendition	537
displayAttachments.....	537
displayEmailAddress.....	537
resolveDistributionList	538
duplicateKey	538
Troubleshooting	539
Messages Cannot Be Imported.....	539
Query Translation.....	539
Attributes	539
Searchable Attributes	539
Returned Attributes.....	540
Operators.....	541
Microsoft SharePoint adapter installation.....	541
Introduction.....	541
Installing the Adapter	541
Installation Requirements	542

Configuring Microsoft SharePoint Server	542
Installing the Adapter Bundle.....	542
Testing Queries	543
Adapter Properties	543
Creating a Backend.....	543
Configurable Properties	544
Mandatory properties	544
bundle	544
host	544
port	544
Optional Properties	544
action.....	545
stopLimit	545
optimizedStopLimit.....	545
protocol	545
supportsLogin	545
loginName	546
loginPassword.....	546
authentication	546
kerberos.env.krb5.conf	547
kerberos.env.login.config	547
proxySet	547
proxyHost.....	547
proxyPort.....	548
dateFormat	548
query	548
queryMethod	548
result	548
trusted.....	549
scope	549
queryExtra	549
attributes	549
language	549
mapin.<SharePoint attributes>	550
mapout.<FS2 attributes>.....	550
createTraceFiles	551
traceFilesPath.....	551
Troubleshooting	551
Internet Information Services Settings	551
Kerberos Authentication Settings.....	552
System.ArgumentNullException Error: misconfiguration	552
Out of Memory Error: configuration limitation.....	552
Query error: host definition	553
Network error: host invalid.....	553
Login error: credentials invalid	554
Query Translation.....	554
Constraints operators.....	554
AND.....	554
OR.....	554
ANDNOT	555
NEAR.....	555
Wildcard.....	555
Operators.....	555
Twitter adapter installation	556
Introduction.....	556
Installing the Adapter	556
Installation Requirements	556
Configuring Microsoft SharePoint Server	557

	Installing the Adapter Bundle.....	557
	Preparing Twitter Account	557
	Adapter Properties	557
	Mandatory Properties	557
	oauth.consumerKey	558
	oauth.consumerSecret.....	558
	Optional Properties	558
	language	558
	compoundScore	558
Chapter 9	REST Services	561
	Introduction.....	561
	DFC Configuration	561
	Docbroker and Global Registry Properties.....	561
	Apache Tomcat.....	562
	VMware vFabric tc Server	562
	Oracle WebLogic Server	562
	IBM WebSphere.....	563
	Red Hat JBoss EAP	564
	Deploy Documentum Platform REST Services on Red Hat JBoss EAP:	564
	Deploy and Configure REST Services on a Docker Environment	565
	Validating REST Deployment	565
Chapter 10	Thumbnail Server	567
	Introduction.....	567
	Thumbnails and Thumbnail Server	567
	Requesting thumbnails	568
	Serving default thumbnails	569
	Installation Overview	569
	Pre-Installation Requirements and Tasks	570
	Setup verification.....	570
	Identifying a Connection Broker in dfc.properties.....	570
	Checking Connection Broker and repository services	571
	Downloading installer	571
	Installing, configuring, and uninstalling Thumbnail Server	571
	Required Thumbnail Server installation information.....	571
	Installing and configuring Thumbnail Server on Windows	572
	Installing and configuring Thumbnail Server on non-Windows	574
	Installing and configuring Thumbnail Server in silent mode	576
	Installing and configuring Thumbnail Server on Docker environment	577
	Configuring Thumbnail Server for SSL.....	578
	Unconfiguration of Thumbnail Server	579
	Uninstalling Thumbnail Server from a Windows host	580
	Uninstalling Thumbnail Server from a non-Windows host	580
	Stopping and starting Thumbnail Server on non-Windows.....	581
	Verifying the Thumbnail Server installation.....	581
	Verifying that Thumbnail Server is running.....	582
	Configuring Thumbnail Server in a trusted content store.....	582
	Administration and Configuration.....	582
	Understanding default thumbnails	583
	Adding default thumbnails	584
	Changing the ticket time-out value	584

	Activating thumbnail logging.....	585
Chapter 11	XML Store	587
	Introduction.....	587
	XML Store enabled repository	588
	XML Store deployment modes	588
	Pre-deployment tasks	590
	Deploying XML Store	590
	XML Store deployment workflow	590
	Entering XML Store license key	591
	Enabling XML Store for a repository	592
	Enabling XML Store to work with multiple Content Servers	593

List of Figures

Figure 1.	Content Server and repository	80
Figure 2.	Basic installation model	82
Figure 3.	Single host Content Servers and multiple repositories	113
Figure 4.	Multiple Content Servers on separate hosts for one repository	114
Figure 5.	Multiple Content Servers on one host for one repository	114
Figure 6.	Content Server and two or more Java Method Servers on a single host	159
Figure 7.	JMS for HA on multiple hosts	160
Figure 8.	Active/passive cluster	162
Figure 9.	Active/active cluster.....	163
Figure 10.	Alternative 1: BOCS Servers at Remote Sites Communicating with Primary Site	222
Figure 11.	Alternative 2: Remote sites, without BOCS servers, using primary site's ACS server.....	223
Figure 12.	The two alternatives for single model 1 combined.....	224
Figure 13.	Single Model 2: Single repository with a distributed storage area	225
Figure 14.	Simple example of distributed architecture.....	234
Figure 15.	Object replication model architecture.....	249
Figure 16.	Federation Model	250
Figure 17.	XYZ jobs	314
Figure 18.	IIS server configuration.....	475
Figure 19.	AppXtender Web Access .NT search	490
Figure 20.	ApplicationXtender user-defined list field	490
Figure 21.	An FS2 query on JDBC	500
Figure 22.	An FS2 query on ODBC	501
Figure 23.	XML Store enabled repository	588
Figure 24.	Embedded XML database	589
Figure 25.	External XML database (xDB).....	589
Figure 26.	XML Store deployment workflow	591

List of Tables

Table 1.	Required information for CIS installation	35
Table 2.	Required environment variable	37
Table 3.	Properties implementing distributed messaging in dmi_queue_item.....	251
Table 4.	Distributed Environment Fields.....	254
Table 5.	DMS Default Directories	257
Table 6.	Starting and Stopping DMS Servers	260
Table 7.	acs.properties keys controlling BOCS cache housekeeping	267
Table 8.	acs.properties keys controlling cache consistency checking.....	268
Table 9.	acs.properties keys controlling URL use.....	269
Table 10.	Properties in ACS configuration objects related to connection broker projection	292
Table 11.	Properties in ACS configuration and server configuration objects related to network proximity values.....	293
Table 12.	dm_PreCacheContent method arguments	297
Table 13.	Example proximity values for a three-site configuration	301
Table 14.	Preferences configuration elements.....	357
Table 15.	Authentication elements (<authentication>).....	366
Table 16.	Environment variables that DFC uses	393
Table 17.	Configuration files for DFC	394
Table 18.	dfc.properties connect and global registry properties	406
Table 19.	dfs-sso-config.properties	409
Table 20.	Authentication Scenarios.....	448
Table 21.	Backend configuration properties — bundle.....	449
Table 22.	Backend configuration properties — host.....	450
Table 23.	Backend configuration properties — port.....	450
Table 24.	Backend configuration properties — protocol.....	450
Table 25.	Backend configuration properties — protocolX.....	450
Table 26.	Backend configuration properties — home.....	450
Table 27.	Backend configuration properties — action	451
Table 28.	Backend configuration properties — actionX	451
Table 29.	Backend configuration properties — method.....	451
Table 30.	Backend configuration properties — methodX.....	451
Table 31.	Backend configuration properties — supportsLogin	452
Table 32.	Backend configuration properties — loginName.....	452
Table 33.	Backend configuration properties — loginPassword	452
Table 34.	Backend configuration properties — proxySet	452
Table 35.	Backend configuration properties — query.....	453

Table 36.	Backend configuration properties — result.....	453
Table 37.	Backend configuration properties — filter	453
Table 38.	Backend configuration properties — stopLimit.....	453
Table 39.	Backend configuration properties — compoundScore	453
Table 40.	Backend configuration properties — expirationTime.....	454
Table 41.	Backend configuration properties — dateFormat	454
Table 42.	Backend configuration properties — duplicate	454
Table 43.	Backend configuration properties — duplicateKey	455
Table 44.	Backend configuration properties — modificationKey.....	455
Table 45.	Backend configuration properties — queryLanguage	456
Table 46.	Backend configuration properties — encoding.....	456
Table 47.	Backend configuration properties — trusted.....	456
Table 48.	Backend configuration properties — supportsSubsumption	457
Table 49.	Backend configuration properties — maxSubsumedQueries.....	457
Table 50.	Backend configuration properties — image	457
Table 51.	Backend configuration properties — client.overview.....	457
Table 52.	Backend configuration properties — client.resultIcon.....	458
Table 53.	Backend configuration properties — client.dfc.types.....	458
Table 54.	Backend configuration properties — zipHTTPResponse.....	458
Table 55.	Backend configuration properties — exposeNativeQuery	459
Table 56.	Backend configuration properties — exposeNativeQueryX	459
Table 57.	Backend configuration properties — authenticationMode	459
Table 58.	Common FS2 attributes — source	460
Table 59.	Common FS2 attributes — score	460
Table 60.	Common FS2 attributes — title	461
Table 61.	Common FS2 attributes — URL.....	461
Table 62.	Common FS2 attributes — date	461
Table 63.	Common FS2 attributes — body	462
Table 64.	Common FS2 attributes — abstract	462
Table 65.	Common FS2 attributes — author.....	462
Table 66.	Common FS2 attributes — keyword	462
Table 67.	Common FS2 attributes — format.....	462
Table 68.	Common FS2 attributes — size	463
Table 69.	Common FS2 attributes — rank.....	463
Table 70.	Common FS2 attributes — site.....	463
Table 71.	Common FS2 attributes — collection.....	463
Table 72.	Documentum adapter properties — bundle.....	466
Table 73.	Documentum adapter properties — baseName	466
Table 74.	Documentum adapter properties — docType	466
Table 75.	Documentum adapter properties — host.....	466
Table 76.	Documentum adapter properties — attributes	467
Table 77.	Documentum adapter properties — client.overview	467
Table 78.	Documentum adapter properties — constraint	467

Table 79.	Documentum adapter properties — optimized	467
Table 80.	Documentum adapter properties — flushResults	467
Table 81.	Documentum adapter properties — stopLimit	468
Table 82.	Documentum adapter properties — optimizedStopLimit	468
Table 83.	Documentum adapter properties — orderClause	468
Table 84.	Documentum adapter properties — preferredRendition	468
Table 85.	Documentum adapter properties — secondaryRendition	469
Table 86.	Documentum adapter properties — dateFormat	469
Table 87.	Documentum adapter properties — viewUrl	469
Table 88.	Documentum adapter properties — filter	469
Table 89.	Documentum adapter properties — useFTI	469
Table 90.	Documentum adapter properties — map.full-text	470
Table 91.	Documentum adapter properties — image	470
Table 92.	Documentum adapter properties — loginName	470
Table 93.	Documentum adapter properties — loginPassword	470
Table 94.	Documentum adapter properties — supportsLogin	471
Table 95.	Documentum adapter properties — mapin.<Documentum attributes>	471
Table 96.	Documentum adapter properties — mapmerge	471
Table 97.	Documentum adapter properties — mapmerge.<FS2 attributes>	472
Table 98.	Documentum adapter properties — mapout.<FS2 attributes>	472
Table 99.	Documentum adapter properties — port	473
Table 100.	Documentum adapter properties — query	473
Table 101.	Documentum adapter properties — result	473
Table 102.	Documentum adapter properties — trusted	473
Table 103.	eRoom adapter properties — bundle	477
Table 104.	eRoom adapter properties — host	477
Table 105.	eRoom adapter properties — action (only ASP-based adapter)	477
Table 106.	eRoom adapter properties — facilityName	477
Table 107.	eRoom adapter properties — eRoomName	478
Table 108.	eRoom adapter properties — createTracefiles (only WS-based adapter)	478
Table 109.	eRoom adapter properties — traceFilePath (only WS-based adapter)	478
Table 110.	eRoom adapter properties — client.overview	479
Table 111.	eRoom adapter properties — filter	479
Table 112.	eRoom adapter properties — image	479
Table 113.	eRoom adapter properties — supportsLogin	479
Table 114.	eRoom adapter properties — loginName	479
Table 115.	eRoom adapter properties — loginPassword	480
Table 116.	eRoom adapter properties — port	480
Table 117.	eRoom adapter properties — protocol	480
Table 118.	eRoom adapter properties — proxySet	480
Table 119.	eRoom adapter properties — query	480
Table 120.	eRoom adapter properties — result	481
Table 121.	eRoom adapter properties — trusted	481

Table 122.	eRoom adapter properties — stopLimit.....	481
Table 123.	eRoom adapter properties — optimizedStopLimit	481
Table 124.	ApplicationXtender adapter properties — action	484
Table 125.	ApplicationXtender adapter properties — dataSourceName.....	484
Table 126.	ApplicationXtender adapter properties — applicationName	484
Table 127.	ApplicationXtender adapter properties — bundle.....	484
Table 128.	ApplicationXtender adapter properties — host.....	485
Table 129.	ApplicationXtender adapter properties — client.overview	485
Table 130.	ApplicationXtender adapter properties — dateFormat	485
Table 131.	ApplicationXtender adapter properties — filter	485
Table 132.	ApplicationXtender adapter properties — image	485
Table 133.	ApplicationXtender adapter properties — supportsLogin	486
Table 134.	ApplicationXtender adapter properties — loginName.....	486
Table 135.	ApplicationXtender adapter properties — loginPassword	486
Table 136.	ApplicationXtender adapter properties — mapin.<ApplicationXtender attributes>	486
Table 137.	ApplicationXtender adapter properties — mapout.<FS2 attributes>	487
Table 138.	ApplicationXtender adapter properties — port.....	487
Table 139.	ApplicationXtender adapter properties — proxySet	487
Table 140.	ApplicationXtender adapter properties — query	487
Table 141.	ApplicationXtender adapter properties — result.....	487
Table 142.	ApplicationXtender adapter properties — stopLimit.....	488
Table 143.	ApplicationXtender adapter properties — optimizedStopLimit	488
Table 144.	ApplicationXtender adapter properties — titleFrom	488
Table 145.	ApplicationXtender adapter properties — trusted	488
Table 146.	ApplicationXtender adapter — Full-text searches	489
Table 147.	ApplicationXtender adapter — Field searches	490
Table 148.	SourceOne adapter properties — bundle.....	493
Table 149.	SourceOne adapter properties — host	493
Table 150.	SourceOne adapter properties — domainName	493
Table 151.	SourceOne adapter properties — searchType.....	494
Table 152.	SourceOne adapter properties — sourceList	494
Table 153.	SourceOne adapter properties — tempFolder	494
Table 154.	SourceOne adapter properties — loginName.....	495
Table 155.	SourceOne adapter properties — loginPassword	495
Table 156.	SourceOne adapter properties — client.overview.....	495
Table 157.	SourceOne adapter properties — image	495
Table 158.	SourceOne adapter properties — dateFormat	495
Table 159.	SourceOne adapter properties — filter	496
Table 160.	SourceOne adapter properties — query.....	496
Table 161.	SourceOne adapter properties — result.....	496
Table 162.	SourceOne adapter properties — trusted.....	496
Table 163.	SourceOne adapter properties — msgRendition	496

Table 164.	SourceOne adapter properties — pageSize	497
Table 165.	SourceOne adapter properties — stopLimit	497
Table 166.	SourceOne adapter properties — protocol	497
Table 167.	SourceOne adapter properties — logonType.....	497
Table 168.	SourceOne constraints operators — AND.....	497
Table 169.	SourceOne constraints operators — OR.....	498
Table 170.	SourceOne constraints operators — ANDNOT	498
Table 171.	SourceOne constraints operators — Phrase.....	498
Table 172.	SourceOne constraints operators — Wildcard.....	498
Table 173.	JDBC/ODBC adapter properties — bundle	503
Table 174.	JDBC/ODBC adapter properties — jdbcDriver	503
Table 175.	JDBC/ODBC adapter properties — jdbcUrl.....	504
Table 176.	JDBC/ODBC adapter properties — mapin.<database attributes>	504
Table 177.	JDBC/ODBC adapter properties — mapout.<FS2 attributes>.....	504
Table 178.	JDBC/ODBC adapter properties — fromClause	505
Table 179.	JDBC/ODBC adapter properties — selectClause	505
Table 180.	JDBC/ODBC adapter properties — client.overview	505
Table 181.	JDBC/ODBC adapter properties — dateOutputFormat.....	505
Table 182.	JDBC/ODBC adapter properties — endClause	506
Table 183.	JDBC/ODBC adapter properties — filter.....	506
Table 184.	JDBC/ODBC adapter properties — useFTI.....	506
Table 185.	JDBC/ODBC adapter properties — map.full-text	506
Table 186.	JDBC/ODBC adapter properties — ignoreCase	507
Table 187.	JDBC/ODBC adapter properties — image.....	507
Table 188.	JDBC/ODBC adapter properties — keyAttribute.....	507
Table 189.	JDBC/ODBC adapter properties — likeMode.....	507
Table 190.	JDBC/ODBC adapter properties — supportsLogin	507
Table 191.	JDBC/ODBC adapter properties — loginName	507
Table 192.	JDBC/ODBC adapter properties — loginPassword.....	508
Table 193.	JDBC/ODBC adapter properties — proxySet	508
Table 194.	JDBC/ODBC adapter properties — query	508
Table 195.	JDBC/ODBC adapter properties — query.enclosingDate	508
Table 196.	JDBC/ODBC adapter properties — result	508
Table 197.	JDBC/ODBC adapter properties — result.enclosingChar	509
Table 198.	JDBC/ODBC adapter properties — useToDate	509
Table 199.	JDBC/ODBC adapter properties — whereClause	509
Table 200.	JDBC/ODBC adapter properties — trusted	509
Table 201.	OpenSearch adapter properties — bundle	511
Table 202.	OpenSearch adapter properties — url	512
Table 203.	OpenSearch adapter properties — image	512
Table 204.	OpenSearch adapter properties — home	512
Table 205.	OpenSearch adapter properties — stopLimit	512

Preface

This guide contains information about installing and configuring Documentum Platform and Platform Extensions products.

Revision History

Revision Date	Description
February 2017	<ul style="list-style-type: none">Updated the Configuring WildFly for SSL, page 144 section.Updated the Configuring IBM WebSphere, page 363 section.
December 2016	Updated for the Language Pack release.
November 2016	<ul style="list-style-type: none">Initial publication.Updated the Tuning PostgreSQL database, page 97 section.

Content Intelligence Services

This chapter provides the instructions for installing the server-side components of Content Intelligence Services (CIS). CIS is administered through Documentum Administrator. The **Documentum Administrator** chapter provides instructions for installing Documentum Administrator.

This chapter is intended primarily for administrators who are installing Content Intelligence Services with Documentum platform or with xCelerated Composition Platform (xCP).

Introduction

EMC Documentum Content Intelligence Services (CIS) is the content analytics component for EMC Documentum. Use content analytics to analyze the textual content of documents and know what the documents are about without having to read them. It enables you to find documents rapidly by enriching search facets with discovered metadata and get the gist of a document by viewing the discovered metadata with the document.

As in the previous versions of Documentum, you can perform categorization for a WDK-based application. This mode of categorization is referred to as classic categorization.

Content Intelligence Services is also available in xCP deployments. The entity detection and pattern detection are only available in xCP applications. Categorization is also available in xCP applications but the taxonomies used for the categorization are not available in the Content Intelligence node in Documentum Administrator.

The *EMC Documentum Content Intelligence Services Administration Guide* provides more information about CIS and the various types of processing.

Components

Content Intelligence Services includes these key components:

- The Content Intelligence Services client (CIS client) creates, manages, or displays the taxonomy used for categorizing documents. Examples of CIS clients are Documentum Administrator, Webtop, xCP 2.1 applications or any custom application using the Content Intelligence Application

Programming Interface (CI API). Use Documentum Administrator to configure CIS. The CI API handles communication between the CIS client, the CIS server, and the Documentum repository.

- The Content Intelligence Services server (CIS server) performs the automatic categorization of documents based on taxonomy and category definitions. It also performs the pattern detection, and triggers the entity detection.
- The entity detection server performs the entity detection analysis using cartridges.
- A repository is also required to store the CIS data (such as taxonomy definitions, document set definitions, and discovered metadata).

Documentum Administrator (DA) includes a Content Intelligence node that enables you to manage CIS resources and the analysis results. You must install Documentum Administrator separately. The **Documentum Administrator** chapter provides instructions for installing Documentum Administrator.

Compatibility

For any Documentum product, EMC recommends you to install CIS with the other Documentum components of the same version. To avoid potential compatibility issues, install CIS 7.3 with Content Server 7.3 and Documentum Administrator 7.3.

CIS 7.3 is compatible with:

- Content Server version 7.1, 7.2, and 7.3.
- Documentum Administrator version 7.1, 7.2, and 7.3.

CIS 7.3 is not compatible with any version of CenterStage or WebPublisher.

CIS does not support Branch Office Caching Services (BOCS).

Preinstallation tasks

- Uninstall any previous installation of CIS. [Uninstalling Content Intelligence Services, page 29](#) provides details to uninstall CIS depending on the installed version.
- Make sure that the free disk space is higher than 4 GB and the temporary directory at least 1 GB before starting the installation. If the temporary directory is not large enough, you can modify the corresponding environment variables (TEMP and TMP) to use another (bigger) directory. You can do it in the Environment variables dialog box on Windows hosts, or using the set command in a command DOS session. The temporary directory can be on a different drive than the one used for the CIS installation.
- If you install CIS on a host machine with an underscore character in its name, the installation does not create the authentication file that stores the credentials for the repository access. To work around this issue, use the IP address of the host machine instead of its name, or enable CIS in DA to trigger the creation of the authentication file.
- If not already done, configure the global registry in Documentum Administrator.

Installation checklist

Prepare the required information

Before starting the installation process, make sure that you know the information required during the installation.

Table 1. Required information for CIS installation

Required information	Description
Host name and port number for the connection broker	The host name of the connection broker and its port number.
Installation Owner Password	The network password for the user performing the installation.
Repository name	The name of the repository that CIS uses.
CIS host	The full name (including the domain name) or IP address of the server on the network (by default, it is set to the machine name).
CIS port	The port of CIS server, default it 8079.
CIS JMX port	The port for CIS JMX agent, default is 8061.
Entity detection server port	The port for the entity detection server, default is 55550. The 20 following ports are reserved and must not be used.
User name and password for CIS repository	The name and password of the user to authenticate CIS server against the repository used by CIS.
Provide the following information about the global registry:	
Repository Name	The name of the global registry.
Global registry user login and password.	The login name and password of the user for the global registry

Make sure that no application (such as an antivirus application) is locking the following ports:

- RMI port 1099, and port range 7130–7229
- RMI port range 55550-55649 (configured during installation)
- port 3690
- port 4445
- port 9000
- port range 10000-15000
- port range 32000-32xxx
- port 40002

The entity detection server uses these ports. If one of them is locked, the installation can continue but you must free the required ports before performing any entity detection analysis. To change the default RMI port, refer to the procedure [Modifying the ports for the entity detection server, page 25](#).

Additional preinstallation tasks for Linux hosts

On Linux hosts, perform the following tasks before you begin the installation process:

- Confirm that these 64-bit RPM packages are installed:
 - glibc-version-release.architecture (e.g. glibc-2.12-1.107.el6.x86_64.rpm)
 - libXau-version-release.architecture (e.g. libXau-1.0.6-4.el6.x86_64.rpm)
 - libxcb-version-release.architecture (e.g. libxcb-1.8.1-1.el6.x86_64.rpm)
 - libX11-version-release.architecture (e.g. libX11-1.5.0-4.el6.x86_64.rpm)
 - libXext-version-release.architecture (e.g. libXext-1.3.1-2.el6.x86_64.rpm)
 - libXi-version-release.architecture (e.g. libXi-1.6.1-3.el6.x86_64.rpm)
 - libXtst-version-release.architecture (e.g. libXtst-1.2.1-2.el6.x86_64.rpm)
- Install these 32-bit RPM packages for Luxid:
 - glibc-version-release.architecture (e.g. glibc-2.12-1.107.el6.i686.rpm)
 - nss-softokn-freebl-version-release.architecture (e.g. nss-softokn-freebl-3.12.9-11.el6.i686.rpm)
 - libgcc-version-release.architecture (e.g. libgcc-4.4.7-3.el6.i686.rpm)
 - libstdc++-version-release.architecture (e.g. libstdc++-4.4.7-3.el6.i686.rpm)

Here version, release, and architecture in the package label are the available version number, release number, and architecture specifier of the packaged software that is compatible with your Linux version.

- Enable the random generator by starting it as root:

```
/sbin/rngd -b -r /dev/urandom -o /dev/random
```
- Set the environment variables manually for CIS.

The environment variables are common to Documentum applications, they are described in the following table. If the installation program does not find the needed environment variables, it aborts the installation.

Some of these environment variables may exist, and if you have other Documentum products installed on your system, some of the values may exist as well. The **Documentum Foundation Classes** chapter provides more information about these variables.

To configure your Linux environment:

1. Set the environment variables:
 - a. Edit the installation owner's .cshrc file (C shell) or .profile file (Bourne or Korn shells). Alternatively, edit a file that has the name .cshrc file or .profile file.
 - b. Add the following variable:

Table 2. Required environment variable

Environment variable	Description
DOCUMENTUM	The full path of the destination directory (and Content Server root, if it is installed on the application server host).

- c. Check the installation requirements:
- /usr/dt/bin and /usr/openwin/bin are on the path
 - DISPLAY is set to localhost:0.0

Additional preinstallation tasks for Windows hosts

Ensure that the file msvcrt71.dll is located at: C:\Windows\SysWOW64 in the target environment before you begin the installation process. The installation process fails if you try to install CIS in an environment that does not include this library.

Migrating status information for document sets

If you upgrade from CIS version 6.0, version 6.0 SP1, or version 6.5, you can use the migration script (introduced in CIS 6.5 SP2). The script enables you to migrate the information related to the status of the document sets already processed for categorization by a previous CIS version. Migrating this information avoids the reprocessing of all documents by the new CIS server.

Note: This procedure is not applicable for entity detection.

To migrate status information:

1. Before installing CIS, create a backup copy of the following file:

```
<CIS installation directory>\deploy\cis.ear\cis.war\repodata\docstatus\document_processing_status.serialized
```

In previous versions of CIS, the default installation directory for CIS was:
C:\Documentum\jboss4.3.0\server\DctmServer_CIS.
2. Install the new version of CIS.
3. Copy the document_processing_status.serialized file to the new CIS server.
4. On CIS host machine, locate the import_docstatus.bat file (on Windows hosts, or import_docstatus on Linux hosts); it can be found at <CIS installation directory>/bin.
5. Run the migration script with the following parameter:

```
import_docstatus.bat document_processing_status.serialized
```

The script output indicates which document sets have been successfully migrated and which ones have been skipped.

Installing Content Intelligence Services

Before performing the following procedure, review the [EMC E-LAB Interoperability Navigator](#) and the product Release Notes to ensure that you have met the hardware and software requirements.

To install Content Intelligence Services:

1. Log in to the CIS server host machine. You must have Administrator privileges on the host machine to run the installation program. The person who installs CIS server is automatically the installation owner.

On Windows, log in as a user with Administrator privileges.

On Linux, you can be a non-superuser to be able to install CIS.

2. From the EMC Online Support (<https://support.emc.com>), download the CIS software file: Content_Intelligence_Services_<version_number>_Windows.zip or Content_Intelligence_Services_<version_number>_<Linux.tar to a temporary directory on the host machine>.

Starter taxonomies are also available from the Support site. The *EMC Documentum Content Server Administration and Configuration Guide* provides more details.

3. Unzip the downloaded file.
4. Run the installer file:
 - On Windows hosts: cisSetup.exe
 - On Linux hosts: cisSetup.bin

On Windows hosts, right-click the installer file cisSetup.exe and select **Run as administrator**.

The Welcome window of the installation wizard appears with a list of products and components that you can install on the machine. If an older version of a product or component is already installed, uninstall it before you proceed.

5. Click **Next**.

The license agreement appears.
6. Read the license agreement, select the option I accept the terms of the license agreement and click **Next**.
7. On Windows hosts, enter **Installation Owner Password**. For the installation owner password, enter the network password for the user performing the installation. The password is required for setting up server security and services on the server that hosts CIS. Click **Next**.

On Linux hosts, this step is skipped.

8. On Windows hosts, specify the destination directory for CIS, and click **Next**.

Documentum products use this directory to store working files, as well as program settings and log files. If the installation program finds a registry entry that contains the required information for a previously installed DFC runtime environment, it skips this step.

On Linux hosts, this step is skipped because you specified the DOCUMENTUM environment variable before the installation.

9. Provide the following information to configure the CIS server:

Field	Description
Repository name	The name of the repository that CIS uses.
CIS host	The full server name, including the name of the machine and the domain name (sub-domain is not supported), or IP address of the host machine for CIS server. By default, it is set to the machine name. You can configure the same host as the production server and as the test server. CIS clients such as Documentum Administrator use the host name to connect to the CIS server. You can modify it later in Documentum Administrator.
CIS port	The port number for CIS server. Default is 8079.
CIS JMX port	The port number for CIS JMX agent. Default is 8061.
Entity detection server host port	The port used for the communication with the entity detection server. The 20 following ports are reserved and must not be used. Default is 55550.

After the installation, you can modify the CIS port and CIS JMX port in the `cis.properties` configuration file as described in *EMC Documentum Content Intelligence Services Administration Guide*.

10. Review the summary. The installation program summarizes what it plans to install and where it plans to install it. Click **Back** to make changes, if any. Otherwise, click **Install**.

The installation checks whether the required ports are available. If some of them are already used, a warning message appears. You can either:

- **Ignore and continue:** Allows you to proceed with the installation but you can modify the ports manually afterwards.
- **Cancel and free ports:** Allows you to cancel the installation, manually free up the required ports, and restart the installation.

If all ports are available, CIS and related products are installed. You cannot undo this installation step.

11. Specify the hostname and port number for the machine that hosts the primary connection broker, and click **Next**.

You can use an IP address or a DNS name.

After the installation, you can change the connection broker host and port values by editing the `dfc.properties` file in `<CIS installation directory>/config/` and modifying the parameters: `dfc.docbroker.host` and `dfc.docbroker.port`.

12. In the Designate Global Registry window, complete these substeps:
 - a. To designate a global registry at a later time, unselect Designate the global registry repository to use and click Next to bypass the following substeps.

You will have to enable a global registry after completing the installation process.

- b. In **Repository Name**, type the name of the repository to be used as the global registry.
- c. In the remaining two text boxes, specify the global registry user login and password. The global registry user must be a user who is restricted to READ privileges on the /System/Modules folder on the repository designated as the global registry.
- d. If the global registry or the global registry user is not configured or inaccessible to the client where you are installing, unselect **Test Connection**.
- e. Click **Next**.

If **Test Connection** is checked, the installation program tries to validate the global registry and user settings that you have specified.

If the installation program detects a global registry on the machine, it skips this step.

13. For the repository used by CIS, specify the following information:

Field	Description
Repository user name for CIS	The name of the user to authenticate the CIS server against the repository used by CIS.
Repository user password for CIS	The password of the user to authenticate the CIS server against the repository.
Repository user domain for CIS	The Windows domain as part of the credentials of the user to authenticate the CIS server against the repository.
Test Connection	Select this option to test the connection with the repository.

The information you provide during this step enables the repository for CIS and deploys the CIS artifacts (DAR file) in the repository. The repository must be running for this action to occur.

If the repository is not running, or if the enabling fails during the installation, you can perform or modify this action later in Documentum Administrator.

If the repository has already been enabled for another CIS instance, the configuration is not modified. A warning message prompts you to modify the configuration in Documentum Administrator.

If CIS artifacts (DAR file) cannot be deployed during CIS installation, deploy the DAR file manually as described in [Deploying CIS artifacts \(DAR file\) manually, page 18](#).

14. Click **Done**.
15. On Linux, if you have not logged in as the root user, change to the root user and then run the command `$DOCUMENTUM/CIS/service/cis_service_register add` to create the CIS service.

By default, CIS is installed in the directory:

- C:\Documentum\CIS on Windows hosts
- \$DOCUMENTUM/cis on Linux hosts

This directory is referenced in CIS documentation as the path <CIS installation directory>.

The default installation folder for the entity detection server is:

- C:\Documentum\CIS\Temis\Luxid on Windows hosts
- \$DOCUMENTUM/cis/Temis/Luxid on Linux hosts

CIS also installs the following third-party software:

- Oracle Outside In Content Access
- Snowball Stemmer Libraries
- Temis Luxid® Annotation Factory and TM360

Check the installation log file `install.log` located in the CIS installation folder to make sure that the installation is successful.

You can modify some installation parameters such as CIS port, CIS JMX port, the repository name for CIS, and so on in the `cis.properties` file as described in *EMC Documentum Content Intelligence Services Administration Guide*.

Installing CIS in silent mode

You can install Content Intelligence Services using the silent (unattended) installation process. You must use this procedure cautiously and only if you cannot use the graphical installation process (installation wizard) because any error is hard to fix during silent installation.

Most parameters correspond to the information asked during the installation using the wizard, refer to [Installing Content Intelligence Services, page 11](#), and write down all parameters.

The silent installation invokes the installation program from a command line and gives it a configuration file that enables the installation to proceed without further interaction.

To install CIS in silent mode:

1. Create a configuration file. You have two possibilities:
 - Update the sample configuration file `silent_install_sample.ini`. The sample file is located at the root of the archive file for the CIS software. For example, `Content_Intelligence_Services_<version_number>_Linux.tar`, at the same level as the installer file.
 - Create the configuration file from scratch by running the installation in real-time, as described in the following step.

To create a configuration file by recording a graphical installation, use a command such as the following at the command prompt:

```
cisSetup.bin -r home/installer/myFile.ini
```

This is the command for Linux. For Windows operating systems, replace `cisSetup.bin` with `cisSetup.exe`.

After the file is generated, you can open the file with an editor, update the file, and change the values of the required variables.

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

Running this command creates `myFile.ini` as an installer configuration file. It runs the installation program interactively and saves your inputs.

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

2. Run the installation program silently by using a command such as the following at the command prompt:

```
cisSetup.bin -f C:\silent_install_sample.ini
```

Note: To run the installer in silent mode, you have to update the value of the variable `INSTALLER_UI` to 'Silent'.

Check the installation log files located in the CIS installation folder to ensure that there are no installation errors.

Installing multiple CIS instances

To install instances of CIS on multiple servers, do the following:

1. Install Content Intelligence Services in normal mode.
2. In the `$Documentum/CIS/config/cis.properties` file on each server, add the property `cis.server.instances.index` and set the value as index of the server. For example, with five CIS servers, the values should be 0, 1, 2, 3, 4 on each server. The property of the first server is `cis.server.instances.index=0`
Each CIS server processes a subset of the documents based on the `objectId` hash.
3. Start multiple CIS servers.
4. Update the latest `ci.jar` in xCP 2.2/2.3 applications and Documentum Administrator version 7.2 or earlier from CIS server `$Documentum/CIS/lib`.
5. Log into Documentum Administrator, navigate to Administrator and click **Configure** at Content Intelligence in the Repository section.
6. Specify multiple production server addresses (hostname or IP, separated by comma) and only one test server address.
Note: For Content Server version 7.2 or earlier, a maximum of 80 characters can be entered for production servers. To support more CIS instances, upgrade to the latest version.
7. Enter your repository credentials to complete the configuration.

Setting up Docker for CIS

Installing and configuring CIS on Ubuntu Docker container

Prerequisites

1. Install the Docker engine.

For example, take CentOS users. Run the following command to install the Docker engine:

```
$ yum install docker-engine
```

Note: The system must have 64-bit architecture and meet a minimum kernel version of 3.10.

Start the Docker engine service if it is not running.

```
$ service docker start
```

For more information about Docker Engine Installation, refer to *Docker Documentation*.

2. Pull the Docker image.

```
$ docker pull ubuntu
```

3. Run the container and modify the ports as per your requirement.

```
$ docker run -it -h HOSTNAME -p 8061:8061 -p 8079:8079 -name cis ubuntu
```

For more details, refer to *Docker Documentation*.

4. Install dependency packages.

Inside the container, run the following command:

```
docker$ apt-get update && apt-get install -y libc6-i386 lib32gcc1 lib32stdc++6
```

This command installs three dependencies for Luxid on the container.

To support UTF-8, run the following command inside the Docker container:

```
docker$ export LANG=en_US.UTF-8
docker$ locale-gen en_US.UTF-8
```

Install the VIM tools with the following command:

```
docker$ apt-get install -y vim
```

5. Copy installer packages.

Use the following command to copy the CIS binaries into a directory (for example, /home) in a Docker container:

```
$ docker cp <path-to-build-CIS> <container-name>:/home
```

Configuring CIS on Docker container

1. Set the environment variable for DOCUMENTUM inside the container:

```
docker$ export DOCUMENTUM=/root/dctm
```

2. Modify the random number generator.

```
docker$ ln -sf urandom /dev/random
```

3. Create the silent install configuration file.

CIS is installed in the Docker container in the silent installation mode.

Use the `silent.ini` file with the following details of all fields to perform the silent installation:

- `INSTALLER_UI`: Set it to silent.
- `CIS.SKIP_DEPLOYING_DAR`: Set to true if it needs to skip creating objects on Content Server. Otherwise, set it to false.
- `CIS_SECTION.CIS_REPOSITORY_NAME`: The name of the repository that CIS uses.
- `CIS_SECTION.CIS_REPOSITORY_USER`: Specify the username for the repository.

- CIS_SECTION.SECURE.CIS_REPOSITORY_PASSWORD: Specify the password for the repository.
- CIS_SECTION.CIS_REPOSITORY_DOMAIN: Specify the user domain for the repository. You can keep this field as empty if no domain is available.
- CIS_SECTION.CIS_HOST: The **hostname** of this Docker container, or the **IP address** of the Docker-engine host.
- CIS_SECTION.CIS_PORT: Set it to 8079.
- CIS_SECTION.CIS_JMX_AGENT_PORT: Set it to 8061.
- CIS_SECTION.LUXID_PORT: Set it to 55550.
- DFC.DOCBROKER_HOST: Specify the **hostname** or **IP address** for the broker host.
- DFC.DOCBROKER_PORT: Specify the port for the broker.
- DFC.DFC_BOF_GLOBAL_REGISTRY_REPOSITORY: The name of the repository to be used as the global registry.
- DFC.DFC_BOF_GLOBAL_REGISTRY_USERNAME: The username for the global registry repository.
- DFC.SECURE.DFC_BOF_GLOBAL_REGISTRY_PASSWORD: The password for the global registry repository.
- USE_CERTIFICATES: Set to false if you do not want to use the user certificates. Otherwise, set it to true.
- DFC_SSL_TRUSTSTORE: Specify a path to the trust store if USE_CERTIFICATES is set to true.
- DFC_SSL_TRUSTSTORE_PASSWORD: Specify the password for the trust store if USE_CERTIFICATES is set to true.
- DFC_SSL_USE_EXISTING_TRUSTSTORE: Set to true to use Java Key Store instead of user-defined trust store. Otherwise, set it to false. The default value is false.

Installing CIS on Docker container

1. Start the silent installation.

Assume the CIS binaries are in */home/cisInstaller/*. Change the permissions for the *cisSetup.bin*:

```
docker$ chmod 755 /home/cisInstaller/cisSetup.bin
```

Access the directory that contains the CIS binaries, and execute the *cisSetup.bin* with *silent.ini*:

```
docker$ cd /home/cisInstaller
docker$ ./cisSetup.bin -f silent.ini
```

2. Start and stop the services.

The services will automatically start after installation is completed.

To stop the CIS service manually, execute the following script file:

```
docker$ cd $DOCUMENTUM/CIS/service
docker$ ./stopCIS
```

To start CIS services manually, run the following script file in the same directory:

```
docker$ ./startCIS
```

Common Notes for Docker environment

Note the following:

- It is recommended to externalize the configuration files from the CIS container to the host machine, to ensure the configuration files are persisted.

Following is a sample command that enables you to run the Docker container and mount volumes that contain necessary files:

```
$ docker run -it -h cis -p 8061:8061 -p 8079:8079 \
-v /home/cisStorage/config:/root/dctm/CIS/config \
-v /home/cisStorage/logs:/root/dctm/CIS/logs \
-v /home/cisStorage/docexclusion:/root/dctm/CIS/repodata/docexclusion \
--name cis ubuntu
```

This command mounts three directories on the host machine to the new created Docker container.

- `/root/dctm/CIS/config` contains the required configuration files
- `/root/dctm/CIS/logs` contains the logs
- `/root/dctm/CIS/repodata/docexclusion` contains the excluded documents

- The CIS server information on Documentum Administrator may not be updated while reinstalling CIS on a specific repository. On Documentum Administrator, navigate to the **Administration > System Information** page, click **Configure**. The CIS server information is displayed on a page as **Production Server** and **Test Server**. Make sure the hostname or the IP addresses are correct.

If the container's hostname is set in the fields, add an entry in the **hosts** file (for example, `./etc/hosts`) on the system where Documentum Administrator is installed.

Example:

```
192.168.0.11    cis
<IP-address-of-host-of-docker-engine> <hostname-of-the-CIS-container>
```

Otherwise, the IP addresses of the Docker-engine host should be set in the fields.

- The [Verifying the deployment of CIS artifacts \(DAR file\)](#), [page 21](#) section describes the steps to verify the installation.

Sample of silent.ini for installing CIS

Assume that the container hostname is **cis**. In this scenario, a complete *silent.ini* file must contain the following fields:

```
INSTALLER_UI=silent
CIS.SKIP_DEPLOYING_DAR=false
CIS_SECTION.CIS_REPOSITORY_NAME=sampleRepository
CIS_SECTION.CIS_REPOSITORY_USER=Administrator
CIS_SECTION.SECURE.CIS_REPOSITORY_PASSWORD=password
CIS_SECTION.CIS_REPOSITORY_DOMAIN=
```

```
CIS_SECTION.CIS_HOST=cis
CIS_SECTION.CIS_PORT=8079
CIS_SECTION.CIS_JMX_AGENT_PORT=8061
CIS_SECTION.LUXID_PORT=55550
DFC.DOCBROKER_HOST=192.168.10.20
DFC.DOCBROKER_PORT=1489
DFC.DFC_BOF_GLOBAL_REGISTRY_REPOSITORY=sampleRepository
DFC.DFC_BOF_GLOBAL_REGISTRY_USERNAME=dm_bof_registry
DFC.SECURE.DFC_BOF_GLOBAL_REGISTRY_PASSWORD=password
USE_CERTIFICATES=false
DFC_SSL_TRUSTSTORE=
DFC_SSL_TRUSTSTORE_PASSWORD=
DFC_SSL_USE_EXISTING_TRUSTSTORE=false
```

Completing the Installation

Authenticated connection between CIS and the repository

The CIS server processes documents in the repository that you have enabled for CIS. CIS works only with one repository. You can enable only one repository with one server in the production mode and one server in the test mode. The test mode is used only for classic categorization processing.

The connection between CIS and the repository is secured with an authentication mechanism when you install CIS, or when you enable the repository for CIS, provide the user name and password to authenticate the CIS server against the repository.

The authentication against the repository is required for CIS to retrieve documents, assign documents to categories, and store the discovered metadata values.

When the CIS server starts, it checks the user credentials against the repository before it opens a session. If the CIS server does not find any credentials, or if the credentials are invalid (for example, after a repository change), it starts in a restricted mode. This mode allows only receiving new or updated credentials. You cannot launch any categorization run but you can change the credentials in Documentum Administrator. When the CIS server receives the valid credentials, it tries to connect to the repository. If successful, it switches to full mode. The section *Defining Content Intelligence Services configuration of the EMC Documentum Content Intelligence Services Administration Guide* provides information on setting the user credentials.

If you modify the authentication information in Documentum Administrator after enabling the repository, it creates another authentication file. There is one authentication file per repository on the CIS server.

The authentication file name is user_<repository_name>.properties and it is available at:

```
<CIS installation directory>/repdata/authentication
```

Deploying CIS artifacts (DAR file) manually

The CIS DAR file (cis_artifacts.dar) is automatically deployed during CIS installation. If the automatic deployment fails, install CIS DAR manually.

If you decide to use CIS with a different repository, install CIS DAR manually.

To install CIS artifacts manually:

1. From the EMC Online Support (<https://support.emc.com>), download the CIS software file: Content_Intelligence_Services_<version_number>_Windows.zip or Content_Intelligence_Services_<version_number>_Linux.tar to a temporary directory on the host machine.
2. Unzip the archive file.
3. Go to the dar folder. This folder contains only one file: cis_artifacts.dar.
4. Install the DAR file using the DAR Installer or headless Composer. The *EMC Documentum Composer User Guide* provides more details.
5. Check that the DAR is installed successfully as described in [Verifying the deployment of CIS artifacts \(DAR file\), page 21](#).

Enabling the repository for CIS

You must enable the repository for CIS use in the following cases:

- The repository has not been enabled for CIS during CIS installation. For example, if the repository was not started or was not reachable.
- You want to use CIS with a different repository which has never been enabled for CIS.

To enable CIS in Documentum Administrator:

1. While CIS server is running, log in to Documentum Administrator.
2. Navigate to Administration > Content Intelligence for the repository you want to process documents from.

3. Click the Enable repository for category assignments link.

The Enable Repository for Content Intelligence page appears.

When you create taxonomies and categories, Documentum Administrator creates corresponding folders, one folder for each taxonomy and category with the same hierarchical relationships.

When the Link to Folders option is active, CIS links categorized documents into the folders corresponding to their assigned categories.

The default location for these folders is in a cabinet named Categories.

The default path for the Content Intelligence administrative information is /System/Application/CI.

You cannot modify these two locations.

4. Specify the following information:

Field	Description
Production Server	The host name of the CIS server in the production mode. [1]
Test Server	For classic categorization only, the host name of CIS server in test mode. [1] [2]
Link assigned documents into category folders and Update document attributes with category assignments	For classic categorization only. The <i>EMC Documentum Content Intelligence Services Administration Guide</i> and <i>EMC Documentum Content Server Administration and Configuration Guide</i> provide more details about these features.
User Name for CIS Server and Password	Name and password of the user to authenticate the CIS server against the repository. The authentication against the repository is required when retrieving documents and assigning documents to categories.

[1] The host name is made of the IP address or the DNS name followed by the port number. The port number is optional. Specify it if the version of Documentum Administrator is 6.0 SP1, 6.5, or 6.5 SP1, or if you have modified the port during installation.

192.168.1.250:8079

The default port number is 8079.

You can define the host names using the IPv6 address. When using an IPv6 address, with or without a specific port number, enclose the host name within square brackets. For example:

[2001:0db8:0:0:0:0:1428:57ab]

[2001:0db8:0:0:0:0:1428:57ab]:5678

[2] CIS enables you to categorize documents in the production mode or the test mode.

Although you can use the same CIS server for both production and testing, separate servers are recommended for better performance and availability. The specified CIS server must be running when you enable the repository. The test mode is only available for categorization and not for entity detection or pattern detection.

5. Click **OK**.

If the repository is already enabled in Documentum Administrator, updating this configuration creates another authentication file.

[Verifying that the repository is enabled for CIS, page 21](#) describes the procedure to check that the repository is enabled correctly.

Validating the Installation

Verifying the deployment of CIS artifacts (DAR file)

After you deploy the CIS DAR file (cis_artifacts.dar), check that the modules are created in Documentum Administrator.

1. Log in to Documentum Administrator.
2. Navigate to **Cabinets > System > Modules > Aspect** and check that the module `cis_annotation_aspect` is present.
3. Verify that the tables `dm_annotation` and `dm_object_annotations` have been created, as described in [Verifying that the tables are created, page 22](#).

Verifying that the repository is enabled for CIS

When you enable a repository for CIS, a number of sections and folders are created. You can check their existence to make sure that the repository is enabled successfully.

To check the existence of CIS sections and folders in the repository:

1. Log in to Documentum Administrator.
2. Navigate to the Content Intelligence node and verify that the following sections are present:
 - Taxonomies
 - Category Class
 - Document Set
 - My Categories
3. Navigate to **Cabinets > System > Applications > CI** and verify that the following folders are present:
 - AttributeProcessing
 - Classes
 - Configuration
 - DocsetConfiguration
 - DocumentSets
 - MetadataExtractionRules
 - Runs
 - TaxonomySnapshots
 - XMLTaxonomies

Verifying that the tables are created

The following tables are created in the repository for CIS:

- When you enable the repository, it creates the table `dm_docstatus`.
- When you deploy the CIS DAR file (`cis_artifacts.dar`), it creates the tables `dm_annotation` and `dm_object_annotations`.

To check the existence of the tables:

1. Log in to Documentum Administrator.
2. Select **Tools > DQL Editor**.
3. Run the query to check the existence of the `dm_docstatus` table:

```
Select * from dm_docstatus
```

The result structure must be:

```
st_object_id st_docset_id st_mode st_last_modified st_date
```

4. Run the query to check the existence of the `dm_annotation` table:

```
Select * from dm_annotation
```

The result structure must be:

```
ann_id ann_type ann_value
```

5. Run the query to check the existence of the `dm_object_annotations` table:

```
Select * from dm_object_annotations
```

The result structure must be:

```
ann_object_id ann_index ann_chronicle_id ann_confidence ann_id
```

Verifying the configuration of the entity detection server

CIS server needs to communicate with the entity detection server to start the detection process, and retrieve the entities.

To verify the configuration of the entity detection server:

1. On CIS host, open the configuration file `<CIS installation directory>/config/cis.properties`.
2. Check that the property `cis.entity.luxid.annotation_server.host` indicates the IP address of the entity detection server.

Verifying that all services are started

You can verify that all services for the entity detection server have started.

On Windows hosts, the CIS services are installed in the automatic startup mode. You can make sure that all services are started correctly, and, if not, start them manually or reboot to start them automatically.

To verify the status of the services (Windows hosts):

1. Select **My Computer > Manage > Services and Applications > Services**.
2. Make sure the service **Documentum Content Intelligence Services** is started. If not, start it.
3. For the entity detection analysis, make sure that the following services are started:
 - Documentum CIS Luxid Admin Server
 - Documentum CIS Luxid Xelda MI Server
 - Documentum CIS Luxid IDE Server
 - Documentum CIS Luxid Annotation Server
 - Documentum CIS Luxid Annotation Node
 - Documentum CIS Luxid Tomcat Server
 - Documentum CIS Luxid Starter (optional)

If you want to start them manually, start the Documentum CIS Luxid Starter service first. This service starts the other services in the correct order.

Troubleshooting Installation Issues

Modifying the ports for the entity detection server

The entity detection server requires several ports. If some of them are used when CIS is installed, you can proceed with the installation and free them afterwards. The following procedure describes how to modify the default RMI port: 1099 after the installation process.

To modify the default RMI port (1099):

1. Stop the entity detection server. Go to **Services and Applications > Services**, right-click **Documentum CIS Luxid Starter** and select **Stop**. If the service is not started, select **Start** and then stop the service.

On Linux, run the **LuxidStarterCmd.sh stop** command to stop the service.

2. Navigate to <CIS installation directory>/cis/Temis/Luxid/.
3. Modify the properties files as follows:

File	Properties
/adminserver/admin.properties	<code>com.temis.server.rmi.port = <RMI port></code>
/IDE/IDEServer.properties	<code>com.temis.server.rmi.port = <RMI port></code>

File	Properties
/node/AnnotationNode.properties	<pre> com.temis.server.rmi.port = <RMI port> com.temis.admin.server.host = localhost:<RMI port> com.temis.lan.ideHost = localhost: <RMI port> com.temis.lan.idkHost = localhost: <RMI port> com.temis.lan.idcHost = localhost: <RMI port> com.temis.lan.tmsHost = localhost: <RMI port> </pre>
/server/AnnotationServer.properties	<pre> com.temis.server.rmi.port = <RMI port> com.temis.admin.server.host = localhost:<RMI port> com.temis.las.ideHost = localhost:<RMI port> </pre>

<RMI port> is the new RMI port number that replaces 1099.

- Restart the entity detection server. **Go to Services and Applications > Services**. Right-click **Documentum CIS Luxid Starter** and select **Start**.

On Linux, run the `LuxidStarterCmd.sh start` command to start the service.

Some Luxid services (4/7) are not started

Problem

Only 4 out of 7 Luxid services are started.

The following error is logged in <installation path for the entity extraction server>\IDE\log\ide.log:

```

ERROR [Timer-5] server.remote.RemoteBinder () - 2009-10-02 08:43:18,118 -
Some remote objects are no more bound to the registry on 127.0.0.1:1099. ERROR [Timer-5]
server.remote.RemoteBinder () - 2009-10-02 08:43:18,134 - Unable to export a registry on the
localhost that accepts requests on port 1099.

```

Cause

Luxid runs a Java RMI registry on port 1099. If another program stops the port, the registry is stopped, and Luxid fails.

Resolution

Modify the RMI port as described in [Modifying the ports for the entity detection server, page 25](#).

CIS installer unable to connect to Content Server

Problem

In Windows, the installer cannot connect to Content Server after uninstalling CIS 7.0 and then installing CIS 7.2.

Resolution

Modify the Windows machine before you install CIS 7.3.

To modify the Windows machine:

1. Rename or remove the file `c:\windows\vpd.properties`.
2. Clean the registry in `HKEY_LOCAL_MACHINE\SOFTWARE\Documentum` and `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Documentum`.

Uninstalling Content Intelligence Services

This section contains the procedures for uninstalling Content Intelligence Services.

Uninstalling (Windows hosts)

Before uninstalling CIS, stop the CIS server and the entity detection server as described in the *EMC Documentum Content Intelligence Services Administration Guide*.

To uninstall:

1. Select **Start > Settings > Control Panel > Add/Remove Programs**.
The Add/Remove window appears.
2. In the **Change or Remove Programs** tab, select **Documentum Content Intelligence Services** in the list of software.
3. Click **Change/Remove**.

The entity detection server component 'Luxid' is visible as a separate product in the **Add/Remove** window. Uninstalling CIS automatically uninstalls this component. Do not try to uninstall it separately from CIS.

Uninstalling (Linux hosts)

Before uninstalling CIS, ensure that you run the `$DOCUMENTUM/CIS/service/cis_service_register delete` command to delete the CIS service.

To uninstall:

1. Navigate to the `$DOCUMENTUM/_uninst/cis` directory.
2. Run CIS uninstaller: `uninstall.bin`.

Downgrading Content Intelligence Services

If you plan to install an earlier version of CIS after installing and uninstalling CIS version 7.0 or earlier, you must also uninstall other components that were installed with CIS. If you do not uninstall these components, they are not updated. The following procedure describes which components to uninstall and the required order.

To uninstall embedded components (Windows hosts):

1. Select **Start > Settings > Control Panel > Add/Remove Programs**.
The Add/Remove window appears.
2. Uninstall the embedded components as follows:
 - a. Select **Documentum Service Wrapper** and click **Change/Remove**.
 - b. Select **Documentum DFC Runtime Environment** and click **Change/Remove**.

For CIS versions 7.1, 7.2, and 7.3, all components are uninstalled automatically when you uninstall CIS.

Content Management Interoperability Services

This chapter is for system administrators or programmers who wish to deploy *Documentum Content Management Interoperability Services* (CMIS). It describes how to deploy Documentum CMIS to a supported servlet container, as well as information about configuration of the CMIS server environment.

Note: The deployment instructions in this guide are applicable for CMIS 1.1 specification. If you are using CMIS 1.0 specification, see *EMC Documentum Content Management Interoperability Services 7.2 Deployment Guide* which contains the deployment instructions.

Introduction

Documentum CMIS is a web application. To deploy Documentum CMIS, you deploy a Documentum CMIS web application archive file to an application server.

Ensure that your system meets the requirements specified in the product Release Notes and [EMC E-LAB Interoperability Navigator](#).

With Documentum CMIS 7.3 you can download Documentum CMIS 7.3 separately for:

- CMIS 1.0 specification
- CMIS 1.1 specification

CMIS 1.1 REST URLs are not backward compatible, if you have not completely applied the REST principles and have hard coded the REST URLs in the client applications. If you have hard coded REST URLs and do not want to upgrade to CMIS 1.1, then you can upgrade to Documentum CMIS 7.3 for CMIS 1.0. Else, you can upgrade to Documentum CMIS 7.3 for CMIS 1.1.

Documentum CMIS 7.3 for CMIS 1.1 still exposes CMIS 1.0 URLs for AtomPub and Web Services. CMIS 1.1 URLs response might contain additional attributes and can break validations at the client applications that were implemented for CMIS 1.0. If you have any validations performed on the CMIS API response and if any of the validations are failing then you can use the CMIS 1.0 URLs. Documentum CMIS 7.3 for CMIS 1.1 is built on Apache Chemistry OpenCMIS Framework and the framework exposes the following URLs:

Browser binding — Example: `http://CMISSERVER:8080/emc-cmis/browser`

RESTful AtomPub Binding 1.0 — Example: `http://CMISSERVER:8080/emc-cmis/resources10`

RESTful AtomPub Binding 1.1 — Example: `http://CMISSERVER:8080/emc-cmis/resources`

Web Services Binding 1.0 — Example: `http://CMISSERVER:8080/emc-cmis/services10/cmis?wsdl`

Web Services Binding 1.1 — Example: `http://CMISSERVER:8080/emc-cmis/services/cmis?wsdl`

You must enable Kerberos SSO before deploying the Documentum CMIS web application.

Note: You cannot use Kerberos SSO at the same time as:

- For SOAP binding, WS-Security UsernameToken Profile 1.1
- For AtomPub binding, HTTP basic authentication
- For Browser binding, HTTP basic authentication

Configuration settings

This section provides information on configuration settings that affect Documentum CMIS, including JVM, Linux, and application properties settings.

General JVM Configuration Settings

To provide adequate heap space and PermGen space for the Documentum CMIS web application, we recommend the following JVM settings:

- `-Xms512m`
- `-Xmx512m`
- `-XX:MaxPermSize=128m`

Using urandom Generators on Linux Systems

There are issues with implementation of pseudo-random number generators on Linux. For more efficient randomization, Linux systems should use urandom generators that are faster but less secure.

To change the source of secure random numbers from random to urandom, set the `java.security.egd` system property as follows:

```
-Djava.security.egd=file:///dev/urandom
```

Specifying this system property will override the `securerandom.source` setting to urandom.

If the application server is on Red Hat Linux, the application server startup script (for example `run.sh` for JBoss, and `startWeblogic.sh` for WebLogic) must be modified to set the option in the JVM.

Documentum CMIS Configuration Files

Documentum CMIS uses these configuration files to set properties for different layers of the application:

- `dfc.properties`, which contains property settings for the underlying DFC (Documentum Foundation Classes) client.
- `cmis-runtime.properties`, which includes properties specific to the Documentum CMIS layer.

DFC Configuration

The `dfc.properties` file provides property settings for the Documentum Foundation Classes runtime. This file is located in `APP-INF/classes` if you are deploying the EAR file, or in `WEB-INF/classes` if you are deploying the WAR file.

The following table describes properties in the `dfc.properties` file that are relevant for Documentum CMIS. For example, the `dfc.properties` file includes the critical settings that are required for Documentum CMIS to reach a connection broker (historically called a *docbroker*) and connect to a Content Server.

Property	Value
<code>dfc.docbroker.host[0]</code>	The fully qualified hostname for the connection broker. You can add backup hosts by adding new properties and incrementing the index number within brackets.
<code>dfc.docbroker.port</code>	If you wish to use a port for the connection broker other than the default of 1489, add a port key.
<code>dfc.globalregistry.repository</code>	The global registry repository name.
<code>dfc.globalregistry.username</code>	The username of the global registry user. The global registry user, who has the default username <code>dm_bof_registry</code> , must have read access to objects in the <code>/System/Modules</code> and <code>/System/NetworkLocations</code> only.
<code>dfc.globalregistry.password</code>	An encrypted password value for the global registry user.
<code>dfc.search.external_sources.enable</code>	<p>True, to enable Documentum Federated Search Services (formerly known as ECIS); false, to disable ECIS.</p> <p>You must specify the Documentum Federated Search Services host machine name in <code>dfc.search.ecis.host</code>.</p>

Property	Value
<code>dfc.search.external_sources.host</code>	Specifies the Documentum Federated Search Services (formerly known as ECIS) host machine name. You must set <code>dfc.search.ecis.enable</code> to <code>true</code> .
<code>dfc.cache.ddinfo.size</code>	Valid values are 1 to 10000. Controls the memory cache size of the Content Server data dictionary. This parameter is required for the CMIS type definition cache.
<code>dfc.cache.type.currency_check_interval</code>	Valid values are 0 to 86400. This parameter is required for the CMIS type definition cache.

You can either copy the username and encrypted password for the global registry user from the `dfc.properties` file on the global registry Content Server host, or you can select another global registry user and encrypt the password using the following command:

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

Documentum CMIS Runtime Properties

The `cmis-runtime.properties` file enables you to set properties that affect application behavior at the CMIS layer.

These properties are optional unless otherwise specified, and if not specified will default to a value documented in the following table. If a supplied value for an integer or Boolean property is invalid, the default value will be used instead.

These items are cached:

- Repository MIME types
- Repository object types
- DFC session service tokens for logged-in users

Name	Description	Default value	Permissible values range
<code>security.configuration.file</code>	Required. File name of security (XWS-Security) configuration for SOAP binding web services.	<code>cmis-security.xml</code> (in <code>cmis-ws-binding.jar</code>)	security file name string

Name	Description	Default value	Permissible values range
cmis.mime_type.cache_expiration_after_x_seconds	<p>Indicates the expiration timeout for MIME type cache.</p> <p>Repository MIME types are cached in memory to help with performance.</p> <p>This property specifies how often the MIME type cache is flushed.</p>	3,600	1 - 8,640,000 (100 days)
cmis.token.cache_expiration_after_x_seconds	<p>Indicates the expiration timeout for service token cache.</p> <p>Service tokens for login users are cached in memory to save the cost of new DFC sessions.</p> <p>This property specifies how often the service token is flushed.</p>	3,600	1 - 8,640,000 (100 days)

Name	Description	Default value	Permissible values range
cmis.type_info.cache_expiration_after_x_seconds	<p>Indicates the expiration timeout (in seconds) for the CMIS type definition cache.</p> <p>When the specified interval has elapsed and if the repository's object types have changed, then the CMIS type definition cache is flushed and reloaded with the updated object types from the repository. All requests that require access to the type definition cache are blocked until the cache is reloaded.</p> <p>The repository's object type definitions are cached in memory to improve performance. In addition, object type and property definitions are loaded into the cache lazily.</p> <p>You might need to tune this value to optimize performance for your deployment.</p>	3,600	1 - 8,640,000 (100 days)
cmis.mime_type.cache_size	<p>The cache size for mime type.</p> <p>The cache size should not be less than the repository list size.</p>	10	1 - 10,000
cmis.token.cache_size	<p>The cache size for service token.</p> <p>The cache size should not be less than the repository list size.</p>	10	1 - 10,000

Name	Description	Default value	Permissible values range
cmis.type_info.cache_size	<p>The cache size for CMIS type definition.</p> <p>The cache size should not be less than the repository list size.</p>	10	1 - 10,000
cmis.default_max_items	<p>The default maximum number of items in a returned collection. This value is used if the client does not provide a value for maxItems.</p> <p>If value = -1 or value = 0 then the value will be set to Integer.MAX_VALUE.</p>	100	-1 - Integer.MAX_VALUE
cmis.max_items_upper_limit	<p>The allowed maximum value for maxItems. This sets an upper limit on maxItems provided by a client.</p> <p>This setting is recommended for system scalability and performance.</p> <p>If value = -1 or value = 0 then the value will be set to Integer.MAX_VALUE.</p>	2,000	-1 - Integer.MAX_VALUE
cmis.exception.full_message.append	<p>Indicates whether to output error messages from layers below CMIS; that is, Documentum error messages.</p> <p>These messages can help to identify the root cause of exceptions.</p>	true	true, false

Name	Description	Default value	Permissible values range
cmis.anonymous_access.repository[index]	<p>The name of the repository to which to grant anonymous access.</p> <p>If one repository is configured as anonymous accessible, set its repository name here. You can set multiple repositories for anonymous access, or set all available repositories to be anonymously accessible.</p>	Not-Set	valid repository name string
cmis.anonymous_access.principal.username[index]	The Documentum login name to be used for anonymous access to the repository with the same index.	Not-Set	valid user login name string
cmis.anonymous_access.principal.password[index]	The Documentum password for the user login with the same index.	Not-Set	valid user password

Anonymous Access Settings

You can configure a principal to allow access to a single repository, to multiple but not all repositories, or to all available repositories.

To make only one repository anonymously accessible, set the `anonymous_access` properties as follows:

```
cmis.anonymous_access.repository[0]=<reponame>
cmis.anonymous_access.principal.username[0]=<username>
cmis.anonymous_access.principal.password[0]=<password>
```

To enable anonymous access to multiple repositories, configure each repository by incrementing the index on the properties:

```
cmis.anonymous_access.repository[0]=<reponame>
cmis.anonymous_access.principal.username[0]=<username>
cmis.anonymous_access.principal.password[0]=<password>
cmis.anonymous_access.repository[1]=<reponame1>
cmis.anonymous_access.principal.username[1]=<username1>
cmis.anonymous_access.principal.password[1]=<password1>
```

If all repositories available to the CMIS services allow anonymous access, and if the username and password for the principal are the same on all repositories, you can use the wildcard, * (asterisk), as follows:

```
cmis.anonymous_access.repository[0]=*
cmis.anonymous_access.principal.username[0]=<username>
cmis.anonymous_access.principal.password[0]=<password>
```

Maximum Items Default and Upper Limit Settings

The CMIS specification defines the `maxItems` parameter as “the maximum number of items to return in a response”. Many CMIS services/resources support this parameter for paging purposes. Typically, a CMIS client will provide a `maxItems` setting in requests to such resources and services. However, in cases when the client does not provide a value for `maxItems`, CMIS will use a default value. The CMIS server administrator can set this default using the `cmis.default_max_items` runtime property.

In some cases a client (perhaps with malicious intent) may set `maxItems` to an excessively large value in a request, which may negatively affect server performance. To guard against this possibility, the CMIS server administrator can set an upper limit to `maxItems` in `cmis.max_items_upper_limit`.

If either property has a value of -1 or 0, CMIS will set no upper bound on the number of items returned, so that the effective limit is `Integer.MAX_VALUE`. CMIS determines the effective `maxItems` value using both of these property settings, as follows:

```
maxItems = MIN(client_or_default_max_items, server_max_items_upper_limit),
where a value of -1 or 0 is treated as equivalent to Integer.MAX_VALUE
```

Configuring Kerberos SSO

EMC Documentum supports Kerberos secure Single-Sign-On (SSO) using Microsoft Active Server Domain Services for Kerberos Key Distribution Center (KDC) services in the following ways:

- In a single domain.
- In two-way trusts between multiple domains in the same forest only; that is, cross-forest trusts are not supported.

Note: In addition, the CMIS client and server must be in the same domain, whereas Content Server can be in a different domain.

To support Kerberos authentication, Documentum CMIS provides server-side JAX-WS handler for SOAP binding and the Servlet filter for AtomPub binding. The Kerberos token is used for authentication, but not for message encryption. Only BASE64 decoding is supported. The full name of EncodingType is:

```
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0
#BASE64Binary
```

Enabling Kerberos SSO

You must enable Kerberos SSO before deploying the Documentum CMIS web application.

Make sure that you have configured the following components:

- (Required for cross-domain support only) Two-way trusts between all applicable domains in the same forest.

Note: In addition, the Documentum CMIS client and Documentum CMIS web application server must be in the same domain, whereas Content Server can be in a different domain.

- Kerberos SSO on Content Server

Note: The *EMC Documentum Content Server Administration and Configuration Guide* provides detailed information.

1. Register the CMIS web application's service principal name (SPN) in the Active Directory and generate a *.keytab file.
2. Enable the application server for Kerberos.

Configuring the Documentum CMIS Web Application's SPN and *.keytab File

To enable authentication of the Documentum CMIS web application on the Kerberos Key Distribution Center (KDC), register the Documentum CMIS web application's service principal name (SPN) on the Active Server KDC using the Microsoft `ktpass` utility. A Kerberos SPN uniquely identifies a service that uses Kerberos authentication. In this case, the service is the Documentum CMIS web application. Executing the `ktpass` utility also generates a *.keytab file. The *.keytab file contains name/value pairs consisting of an SPN and a long-term key derived from a password. Both the Documentum CMIS web application and the KDC must be able to access the *.keytab file. You copy the *.keytab file to the Documentum CMIS web application machine (the machine where the Kerberos service ticket (ST) is validated) and specify the location of the *.keytab file in the JAAS configuration.

Note: Although the *.keytab file is usually used on non-Windows machines, Documentum CMIS leverages the *.keytab file to improve network performance by eliminating Kerberos authentication communication between Windows machines and the KDC.

In some cases, you can register the SPNs of more than one Documentum CMIS web application to the same account. For example, in load-balanced environments support for Kerberos can be achieved by joining all load-balanced nodes into a single account and assigning a single SPN to the cluster. If access to the service is required through a different SPN (for example, based on the service host IP address rather than the load balancer name), then this SPN can also be registered with the same account. The following procedure describes the main steps for registering an SPN using a one-to-one mapping between the Documentum CMIS web application's SPN and user account, or a many-to-one mapping in which multiple SPNs are registered to one user account.

To Configure the SPN and keytab File:

1. Create a user (or use an existing one) for the Documentum CMIS web application in the Active Directory.

Note: Make sure to enable delegation trust for the service accounts who create the SPNs.

2. Map the Documentum CMIS web application's SPN to a user and generate the *.keytab file.

Mapping the SPN to a User Name

The recommended SPN format for a Documentum CMIS web application is:

`HTTP/<host>:<port>@<REALM>`

where:

- <host_name> is the name of the machine on which the Documentum CMIS web application is deployed. EMC recommends using a host name rather than an IP address as the host string. For example, myhost.mydomain.com. realm is the name of the Kerberos realm, which is defined in the Kerberos configuration file.
- <port> is the port at which the Documentum CMIS web application is listening.

Note:

- When using Windows Integrated Security, Internet Explorer uses the HTTP-service-type of SPN to request service tickets and to process requests. Therefore, using the HTTP protocol in the SPN is more appropriate and consistent for both the CMIS SOAP and HTTP protocols.
- By default, Windows Server 2008 R2 SP1 does not support DES-related ciphers (for example, DES-CBC-MD5).
- *Microsoft documentation* contains the information on the ktpass utility syntax.

To map SPN to a user name perform one of the following tasks:

- To map the SPN to a user name using a one-to-one mapping, execute the ktpass utility as follows:

Note: For a one-to-one mapping, do not map the same SPN to more than one user account.

```
ktpass /pass <password> -out <keytab_file> -princ <SPN>
-crypto <crypto_type> +DumpSalt -ptype KRB5_NT_PRINCIPAL +desOnly
/mapOp set /mapUser <user_name> /target <domain_controller>
```

- To map multiple SPNs to a user name using many-to-one mapping, perform the following steps:

1. Execute the ktpass utility as follows:

```
ktpass /pass <password> -out <keytab_file> -princ <SPN>
-crypto <crypto_type> +DumpSalt -ptype KRB5_NT_PRINCIPAL +desOnly /mapOp
set /mapUser <user_name> /target <domain_controller>
```

Remember the salt string and the key version number (vno) because you need to use them in step 3.

2. To map the next SPN to the same user account, execute the setspn utility as follows:

```
setspn -A <SPN> <user_name>
```

3. Execute ktpass utility for the second SPN without setting with the same user as follows:

Note: Use the salt and key version number (kvno) that were displayed as the output in step 1.

```
ktpass /pass <password> -out <keytab_file> -princ <SPN>
-crypto <crypto_type> +DumpSalt -ptype KRB5_NT_PRINCIPAL +desOnly
/mapOp set +RawSalt <salt> -in <keytab_file> -kvno <vno>
```

4. Repeat Steps 2 and 3 for each additional SPN.

Configuring the Application Server for Kerberos

Configuring krb5.ini and cmis-runtime.properties Files

1. For the Documentum CMIS web application to perform Kerberos delegation, set the following properties in `cmis-runtime.properties`:

- For single-domain support:

Property	Description
<code>cmis.spn</code>	The Documentum CMIS web application's SPN. The syntax is: <code>HTTP/<HOSTNAME>:<PORT>@<REALM></code>
<code>cmis.jaas.conf</code>	The path to the <code>jaas.conf</code> file (for example, <code>C:/jaas.conf</code>).
<code>cmis.krb5.conf</code>	The path to the <code>krb5.ini</code> file (for example, <code>C:/Windows/krb5.ini</code>).

- For multi-domain support:

Property	Description
<code>cmis.spn</code>	The Documentum CMIS web application's SPN. The syntax is: <code>HTTP/<HOSTNAME>:<PORT></code>
<code>cmis.jcsi.nameserver</code>	IP addresses for Kerberos name servers.
<code>cmis.jcsi.maxpacketsize</code>	The maximum packet size setting for multi-domain Kerberos support. QUEST libraries use TCP as the default protocol over UDP for communicating with the KDC. It uses Nagle's algorithm when Kerberos requests are small (less than an Ethernet packet size; for example, 1420) and causes a delay. QUEST still supports UDP if you want to use this protocol. Switching from TCP to UDP can be done by setting this property. If the packet size is less than or equal to the value provided in this property, then the QUEST library uses UDP to communicate with the KDC; otherwise, it uses TCP. The value will overwrite the <code>jcsi.kerberos.maxpacketsize</code> system variable. Default is <code>not-set</code> .

2. Create the krb5.ini file as follows:

Note: This file is typically created in C:\Windows.

```
[libdefaults]
default_realm = <REALM>
forwardable = true    ticket_lifetime = 24h
clockskew = 72000
default_tkt_enctypes =
default_tgs_enctypes =

[realms]
<REALM> = {
kdc = <kdc_server_ip>
admin_server = <admin_server_ip>
}

[domain_realm]
<domain> = <REALM>

[logging]
default = c:\kdc.log
kdc = c:\kdc.log

[appdefaults]
autologin = true
forward = true
forwardable = true
encrypt = true
```

<kdc_server_ip>	The IP address of the KDC server.
<admin_server_ip>	The IP address of the Administration server.
<domain>	The domain in which the Documentum CMIS web application's SPN resides.
<REALM>	The realm name. For example: MYDOMAIN.MYCORP.COM

Configuring the JAAS.conf file

An application server's JAAS configuration file specifies properties for the `LoginContext` name, Kerberos login module, the Documentum CMIS web application's SPN, and the location of the `*.keytab` file.

The location and format of the JAAS configuration settings might be different for each application server. Unless otherwise specified in the application server deployment instructions, a configuration file setting can also be specified as follows:

- In `cmis-runtime.properties`
- In a JVM command-line parameter; for example:
`-Djava.security.auth.login.config=<path_to_JAAS.config>`

Example 2-1. Single-Domain JAAS Configuration referring to SUN JDK

```
{
com.sun.security.auth.module.Krb5LoginModule required
```

```
debug=false
principal=<SPN>
refreshKrb5Config=true
useKeyTab=true
storeKey=true
doNotPrompt=true
useTicketCache=false
isInitiator=false
keyTab=<cmisuser_keytab_path>;
};
```

Example 2-2. Single-Domain JAAS Configuration referring to IBM JDK

```
{
  com.ibm.security.auth.module.Krb5LoginModule required
  debug=false
  credsType="both"
  useKeyTab=<cmisuser_keytab_path>
  principal=<SPN>;
};
```

Example 2-3. JAAS Configuration referring to QUEST Libraries which support both Single Domain and Multi Domain

```
{
  com.dstc.security.kerberos.jaas.KerberosLoginModule required
  debug=false
  principal=<SPN>
  realm="CMISKDC.IIG.EMC.COM"
  refreshKrb5Config=true
  noTGT=true
  useKeyTab=true
  storeKey=true
  doNotPrompt=true
  useTicketCache=false
  isInitiator=false
  keyTab=<cmisuser_keytab_path>;
};
```

Note: In WebSphere Application Server, the JAAS configuration must be specified in <WAS_Installation_path>\AppServer\profiles\<APP_SERVER_NODE_NAME>\properties\wsjaas.conf.

<loginContext>	<p>Corresponds to the Documentum CMIS web application's SPN. You replace separator characters with hyphen characters and omit the @REALM segment in the SPN. For example, the following LoginContext is derived from the corresponding SPN:</p> <ul style="list-style-type: none"> • LoginContext: <code>HTTP-myhost-mydomain-com-8080</code> • SPN: <code>HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM</code> <p>Note: Make sure that the SPN in the JAAS configuration matches the SPN defined in <code>cmis-runtime.properties</code>.</p>	
<LoginModule>	Specify the Kerberos login module to be used to perform user authentication.	
	<p>Single Domain:</p> <ul style="list-style-type: none"> • Referring to Sun JDK: <code>com.sun.security.auth.module.Krb5LoginModule</code> • Referring to IBM JDK: <code>com.ibm.security.auth.module.Krb5LoginModule</code> • Referring to QUEST Libraries: <code>com.dstc.security.kerberos.jaas.KerberosLoginModule</code> 	<p>Multi-Domain: <code>com.dstc.security.kerberos.jaas.KerberosLoginModule</code></p>
	<p>Note: For QUEST login modules, if you want to enable ticket cache, perform one of the following operations. Otherwise, disable ticket cache by setting <code>useTicketCache</code> to <code>false</code>.</p> <ul style="list-style-type: none"> • Enable <code>createTicketCache</code>: <code>useTicketCache=true</code> <code>createTicketCache=true</code> • Enable <code>createTicketCache</code> and specify a cache path: <code>useTicketCache=true</code> <code>createTicketCache=true</code> <code>ticketCache=<cache_path></code> 	
<SPN>	<p>The Documentum CMIS web application's SPN.</p> <p>For example, for SUN and IBM login modules: <code>HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM</code></p> <p>For QUEST login modules, the SPN does not contain the @ character and the string after that. For example: <code>HTTP/myhost.mydomain.com:8080</code></p>	

<code><REALM></code>	(Multi-domain support only) The realm name. For example: <code>@MYDOMAIN.MYCORP.COM</code>
<code><cmisuser_keytab_path></code>	The path to the user account's *.keytab file on the Documentum CMIS web application. For example: <code>c:\cmisuser.keytab</code>

Configuring the Documentum CMIS Web Application

Make the following changes to the Documentum CMIS web application and redeploy it.

1. For SOAP binding, change `sun-jaxws.xml` to specify the class in bold:

Note: Make sure that you do not specify the user name `AuthHandler` in addition to the Kerberos `WsSecurityKerberosTokenHandler`.

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>WS-Security Kerberos Token Profile
      1.1</handler-name>
      <handler-class>com.emc.documentum.fs.cmis.impl.
auth.wshandler.WsSecurityKerberosTokenHandler
</handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

2. For AtomPub binding, change `web.xml` to specify the class in bold.

Note: Make sure that you do not specify the HTTP Basic Auth filter in addition to the Kerberos Negotiate Auth filter.

```
<servlet>
  <servlet-name>cmisatom</servlet-name>
  <servlet-class>com.emc.documentum.fs.cmis.impl.atompub.servlet.
EmcCmisAtomPubServlet</servlet-class>
  <!--
    Customizations for 1.1 AtomPub binding:
    1. To enable Kerberos SSO authentication, modify the
    following callContextHandler param by
    replacing com.emc.documentum.fs.cmis.impl.auth.
    callcontexthandler.HttpBasicAuthCallContextHandler
    with com.emc.documentum.fs.cmis.impl.auth.
callcontexthandler.HttpKerberosAuthCallContextHandler
  -->
  <init-param>
    <param-name>callContextHandler</param-name>
  <param-value>
    com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
HttpBasicAuthCallContextHandler
  </param-value>
</init-param>
<init-param>
  <param-name>cmisVersion</param-name>
  <param-value>1.1</param-value>
</init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
```

3. For Browser binding, change `web.xml` to specify the class in bold.

Note: Make sure that you do not specify the HTTP Basic Auth filter in addition to the Kerberos Negotiate Auth filter.

```
<servlet>
    <servlet-name>cmisbrowser</servlet-name>
    <servlet-class>
com.emc.documentum.fs.cmis.impl.browser.servlet.EmcCmisBrowserBindingServlet
</servlet-class>
    <!--
        The default authentication mechanism for browser binding is HTTP basic
        authentication with callContextHandler parameter set to value
        com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
HttpBasicAuthCallContextHandler
        The following customizations are allowed on EMC CMIS Documentum 1.1 for
        Browser binding:
        1. To enable Kerberos SSO authentication, modify the following call
        ContextHandler parameter by replacing com.emc.documentum.fs.cmis.
        impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler with
com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.HttpKerberos
AuthCallContextHandler

        2. To enable HTTP basic with tokens based authentication, modify the
        following callContextHandler parameter by replacing com.emc.documentum.
        fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler with
        com.emc.documentum.fs.cmis.impl.browser.token.impl.ExtendedTokenCall
        ContextHandler

        3. To enable Kerberos SSO with tokens based authentication, modify the
        following callContextHandler parameter by replacing com.emc.documentum.
        fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCallContextHandlerwith
        com.emc.documentum.fs.cmis.impl.browser.token.impl.EmcKerberosTokenCall
        ContextHandler
    -->
    <init-param>
        <param-name>callContextHandler</param-name>
        <param-value>
com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler
</param-value>
        </init-param>
        <load-on-startup>2</load-on-startup>
</servlet>
```

4. Add the following files into the appropriate web archive (either `emc-cmis-weblogic.ear` or `emc-cmis.war\WEB-INF\lib`) as follows:

```
vsj-license.jar (required for multi-domain support)
vsj-standard-3.3.jar (required for multi-domain support)
questFixForJDK7.jar (required for multi-domain support on JDK 7)
util.jar
krbutil.jar
```

Logging for Kerberos

Logging information is recorded in the Documentum CMIS web application's Kerberos-related handlers and filters. To log debug information, enable log4j's DEBUG level in `com.emc.documentum.fs.cmis`. JAAS/GSS-API also has options for enabling Kerberos logging.

- *Oracle documentation* provides more information about enabling JAAS debugging.
- *Oracle documentation* provides more information about Kerberos error codes.

Performance Best Practices

To make CMIS, multi-domain Kerberos perform better, apply the best practices described in this section.

Note: The best practices and/or test results are derived or obtained after testing the product in the EMC testing environment. Every effort is made to simulate common customer usage scenarios during performance testing, but actual performance results will vary due to differences in hardware and software configurations, data, and other variables.

QUEST TCP/UDP Settings

Because of QUEST library's default settings for multi-domain Kerberos support, it takes much longer to acquire a Content Server service ticket than it does to authenticate a login credential. The overhead of QUEST can be over 600 milliseconds, as indicated by the test results in the following table.

Transaction Response Time (Milliseconds)	Multi-Domain	Single Domain
Kerberos Delegate	654	447
DFC getSession	30	29

As many as three requests are sent to KDCs to acquire a service ticket. Although each request's response time is very fast (less than 4 milliseconds), the delay between requests is over 200 milliseconds. This delay occurs when Nagle's algorithm is triggered to combine small segments into a larger one. QUEST sends TCP requests with two segments; however when the segment size is less than one Ethernet packet, Nagle's algorithm is triggered.

To reduce these kinds of delays, set the `maxpacketsize` parameter, which specifies the threshold (in bytes) at which QUEST switches from UDP to TCP, as follows:

```
set maxpacketsize = 2000
```

Note: This setting is consistent with Windows.

Transaction Response Time (Milliseconds)	Multi-Domain	
Single User Test	With QUEST's default settings	With QUEST's tuned settings

Transaction Response Time (Milliseconds)	Multi-Domain	
Kerberos Delegate	654	32
DFC getSession	30	45

Configuring token-based authentication for Browser Binding

When using Browser binding, you can configure CMIS Web Application with token-based authentication using HTTPBasicAuthentication or Kerberos SSO.

HttpBasicAuthentication with Tokens

To configure HttpBasicAuthentication with tokens for Browser binding, change `web.xml` to specify the class in bold.

```
<servlet>
    <servlet-name>cmisbrowser</servlet-name>
    <servlet-class>com.emc.documentum.fs.cmis.impl.browser.servlet.EmcCmis
    BrowserBindingServlet</servlet-class>

    <!--
    The default authentication mechanism for browser binding is HTTP basic
    authentication with callContextHandler parameter set to value
    com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.HttpBasicAuth
    CallContextHandler
    The following customizations are allowed on EMC CMIS Documentum 1.1 for
    Browser binding:

    1. To enable Kerberos SSO authentication, modify the following call
    ContextHandler parameter by replacing com.emc.documentum.fs.cmis.
    impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler
    with com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
    HttpKerberosAuthCallContextHandler

    2. To enable HTTP basic with tokens based authentication, modify the
    following callContextHandler parameter by replacing com.emc.documentum.
    fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler
    with com.emc.documentum.fs.cmis.impl.browser.token.impl.
    ExtendedTokenCallContextHandler

    3. To enable Kerberos SSO with tokens based authentication, modify the
    following callContextHandler parameter by replacing com.emc.documentum.
    fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler
    with com.emc.documentum.fs.cmis.impl.browser.token.impl.EmcKerberos
    TokenCallContextHandler

    -->
    <init-param>
        <param-name>callContextHandler</param-name>
        <param-value>com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
    HttpBasicAuthCallContextHandler</param-value>
    </init-param>
```

```
<load-on-startup>2</load-on-startup>
</servlet>
```

Kerberos SSO with Tokens

To configure Kerberos SSO with token-based authentication for Browser binding, change `web.xml` to specify the class in bold.

```
<servlet>
  <servlet-name>cmisbrowser</servlet-name>
  <servlet-class>com.emc.documentum.fs.cmis.impl.browser.servlet.EmcCmis
    BrowserBindingServlet</servlet-class>

  <!--
    The default authentication mechanism for browser binding is HTTP basic
    authentication with callContextHandler parameter set to value
    com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.HttpBasic
    AuthCallContextHandler
    The following customizations are allowed on EMC CMIS Documentum 1.1 for
    Browser binding:

    1. To enable Kerberos SSO authentication, modify the following call
    ContextHandler parameter by replacing com.emc.documentum.fs.cmis.
    impl.auth.callcontexthandler.HttpBasicAuthCallContextHandler
    with com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
    HttpKerberosAuthCallContextHandler

    2. To enable HTTP basic with tokens based authentication, modify
    the following callContextHandler parameter by replacing com.emc.
    documentum.fs.cmis.impl.auth.callcontexthandler.HttpBasicAuth
    CallContextHandler with com.emc.documentum.fs.cmis.impl.browser.
    token.impl.ExtendedTokenCallContextHandler

    3. To enable Kerberos SSO with tokens based authentication, modify
    the following callContextHandler parameter by replacing com.emc.
    documentum.fs.cmis.impl.auth.callcontexthandler.HttpBasicAuthCall
    ContextHandler with com.emc.documentum.fs.cmis.impl.
    browser.token.impl.EmcKerberosTokenCallContextHandler

  -->
  <init-param>
    <param-name>callContextHandler</param-name>
    <param-value>com.emc.documentum.fs.cmis.impl.auth.callcontexthandler.
    HttpBasicAuthCallContextHandler</param-value>
  </init-param>

  <load-on-startup>2</load-on-startup>
</servlet>
```

Deploying to supported application servers

You can deploy a Documentum CMIS web application archive file to an application server. If you are configuring Kerberos SSO, you must perform steps before deploying the Documentum CMIS web application.

To deploy the Documentum CMIS web application, deploy the appropriate archive file as shown in the following table.

Application Server	Archive File
Apache Tomcat	emc-cmis.war
VMware vFabric tc Server	
IBM WebSphere	
Oracle WebLogic	emc-cmis-weblogic.ear

Apache Tomcat

Make sure that you are deploying CMIS on a certified version of Apache Tomcat. [EMC E-LAB Interoperability Navigator](#) provides detailed information.

Make sure that the Tomcat JVM settings meet the recommendations (the Tomcat default settings may not be adequate). The Apache Tomcat web site provides detailed information.

Copy the WAR file to the `<TomcatHome>/webapps` directory.

Tomcat unpacks the WAR file to the `<TomcatHome>/webapps/<application_name>` directory, where `<application_name>` is the name of the WAR file without the file extension.

VMware vFabric tc Server

Make sure that you are deploying CMIS on a certified version of VMware vFabric tc Server. [EMC E-LAB Interoperability Navigator](#) provides detailed information.

The VMware vFabric tc Server web site provides detailed information about deploying web applications.

Oracle WebLogic Server

Make sure that you are deploying CMIS on a certified version of Oracle WebLogic Server. [EMC E-LAB Interoperability Navigator](#) provides detailed information.

To successfully deploy the Documentum CMIS web application on Oracle WebLogic, disable the container's HTTP Basic authentication.

1. To disable the WebLogic container's HTTP Basic authentication, edit the following file and save it:
`<WebLogic_home>/user_projects/domains/<domain>/config/config.xml`.
 - a. Find the `<security-configuration>` section of the file.
 - b. If `enforce-valid-basic-auth-credentials` is already defined in this section, then change its value to `false`. Otherwise, add the following line before the `</security-configuration>` line:
`<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>`
2. Restart the WebLogic server.
3. Deploy the Documentum CMIS web application's archive file (`emc-cmis-weblogic.ear`) using the WebLogic Console.

For more information about deploying web applications to WebLogic, refer to the Oracle WebLogic Server web site.

IBM WebSphere

Make sure that you are deploying CMIS on a certified version of IBM WebSphere. [EMC E-LAB Interoperability Navigator](#) provides detailed information.

Use the following procedure to deploy CMIS using the Integrated Solutions Console.

1. Start WebSphere server.
2. Add a custom property:
 - a. Select **Application servers > ServerName > Web container > Custom properties**.
 - b. Add a custom property `com.ibm.ws.webcontainer.removetrailingservletpathslash` with the value `true`.
3. Install `emc-cmis.war`.
4. Configure the class loader policy for the CMIS application for WebSphere versions earlier than 8.5:
 - a. Set **Class Loader Order** to **Classes loaded with local class loader first (parent last)**.
 - b. Set **War Class Loader Policy** to **Single class loader for application**.

5. Add the JVM runtime option `-noverify` or `-XX:-UseSplitVerifier` using the WebSphere Admin console, by performing the following steps:
 - a. Select **Servers > Server Types > WebSphere application servers**.
 - b. Click the `<server name>` link.
 - c. In the **Server Infrastructure** area, expand **Java and Process Management** and click the **Process definition** link.
 - d. In the **Additional Properties** area, click the **Java Virtual Machine** link.
 - e. In the **Configuration** tab, specify the `-noverify` value in the **Generic JVM arguments** text box.
 - f. Click **Apply**.
 - g. Click **Save**.
6. Start the CMIS application.

Post deployment

This section describes deployment validation and the CMIS service addresses.

Validation

On successful deployment, you should be able to access the CMIS home page at the following URL:

`http://<host>:<port>/<contextPath>`

Note: The application context path will vary depending on your deployment. In most deployments the default context path is `emc-cmis`, based on the name of the archive file.

RESTful AtomPub Service Document

The service document defining the RESTful AtomPub binding can be obtained from this address:

`http://<host>:<port>/<contextPath>/resources/`

Web Service Entry Points

You can view the WSDL for any of the SOAP web services by using a URL like the following:

`http://<host>:<port>/<contextPath>/services/RepositoryService?wsdl`

The WSDL files for each of the CMIS web services are essentially identical: each one defines endpoints for all of the CMIS web services, which are shown in the following table:

web service	Address
ACLService	http://<host>:<port>/<contextPath>/services/ACLService
DiscoveryService	http://<host>:<port>/<contextPath>/services/DiscoveryService
MultiFilingService	http://<host>:<port>/<contextPath>/services/MultiFilingService
NavigationService	http://<host>:<port>/<contextPath>/services/NavigationService
ObjectService	http://<host>:<port>/<contextPath>/services/ObjectService
RelationshipService	http://<host>:<port>/<contextPath>/services/RelationshipService
RepositoryService	http://<host>:<port>/<contextPath>/services/RepositoryService
VersioningService	http://<host>:<port>/<contextPath>/services/VersioningService

Browser Binding URLs

The document returned by the following Service URL provides the repository information for all available repositories:

`http://<host>:<port>/<contextPath>/browser/`

Each repository information must contain the following two additional properties:

- repositoryUrl: The Repository URL
- rootFolderUrl: The Root Folder URL

Content Server

This chapter is intended for system administrators responsible for installing and configuring Content Server. The *EMC Documentum System Upgrade and Migration Guide* contains the instructions on upgrading Content Server.

Installation overview

Use the information provided in this to understand the Content Server architecture and the components you must install while installing the Content Server. Ensure that you plan the installation based on your business needs.

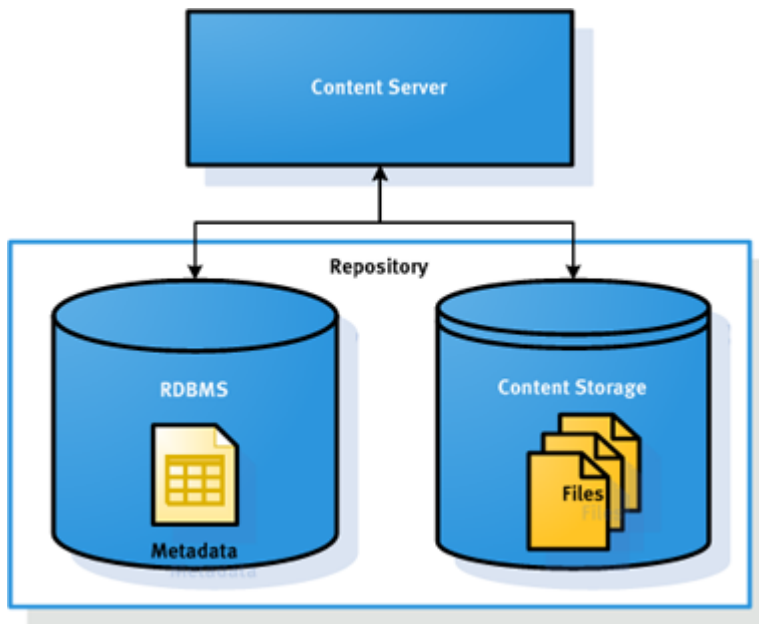
Content Server architecture

This section provides basic information about the Content Server architecture and the components that are installed during the installation process.

Content Server and repository

EMC Documentum Content Server is a powerful, robust, and scalable enterprise content management system that provides advanced content management and process management functions to organize, control, and access all your information assets in your organization.

Content Server manages content stored in object-based repositories. A *repository* comprises of a *storage* that stores native content files, and a *relational database management system (RDBMS)* that stores properties of these content files called *metadata*, such as document owner, version, and creation date. Metadata describes the content object and its relationship between other objects in repository, and is used to manage and search for content.

Figure 1. Content Server and repository

Content Server itself consists of several distinct processes and components that are mostly transparent to the user during installation:

- Application server

Content Server uses a private embedded *application server* as a container for Java Method Server (JMS), Accelerated Content Server (ACS), and other components.

- Java Method Server (JMS)

Java Method Server (JMS) is a customized version of WildFly that executes Content Server Java methods. One Java Method Server is installed with each Content Server installation.

- Accelerated Content Services (ACS) server

Accelerated Content Services (ACS) Server is a lightweight server that is automatically created during Content Server installation. The ACS server reads and writes content for web-based client applications using HTTP and HTTPS protocols. ACS servers do not modify object metadata but write content to storage areas.

- Documentum Foundation Classes (DFC)

Documentum Foundation Classes (DFC) provides the programming interface that client applications use to communicate with Content Server.

Connection broker

Content Server clients connect to Content Server through connection brokers. A *connection broker* is a process that provides client sessions with Content Server connection information, such as their IP addresses and port numbers, as well as proximity values of their network locations.

The connection brokers that handle a client connection request are defined in the `dfc.properties` file of the client. When a user or application requests a repository connection, the request goes to a

connection broker identified in the client `dfc.properties` file. The connection broker returns the connection information for the repository or a particular server identified in the request.

Connection brokers do not request information from Content Servers, but rely on the servers to regularly broadcast their connection information to them. When Content Server starts, it automatically broadcasts information about itself to one or more connection brokers. Each connection broker that receives the broadcast adds the Content Server to its list of available servers. The information on connection broker is configured in the server config object (`dm_server_config`) of the server.

Each Content Server installation must have at least one connection broker. The first connection broker is started as part of the installation process.

When a client application wants to connect to a repository, the following occurs:

1. The client contacts the connection broker and requests the information it needs to connect with a Content Server for the requested repository.
2. The connection broker sends back the IP address for the host on which the Content Server resides and the port number that the Content Server is using.
3. The client application uses that information to open a connection to Content Server.

Global registry

When a Content Server installation includes multiple repositories, certain installation-wide elements are shared among all repositories.

To manage these installation-wide elements, each Content Server installation has a central repository called the *global registry*. You must designate a repository as a global registry, even if you plan to install one repository. The global registry is a repository like any other repository, except that all other repositories connect to it when they need an installation-wide element.

If you have a one-repository implementation, that repository is both a content repository and a global registry. If you have a Content Server implementation larger than a departmental one, consider creating a separate repository and designate that repository to be the global registry only.

Global registry user

A *global registry user* is created in all repositories, regardless of whether the repository is configured as a global registry.

- If you configure a repository as a global registry, you provide the username and password for the global registry user and the user state is set to Active.
- If you do not configure a repository as a global registry, a global registry user is created with the default username `dm_bof_registry` and the user state is set to Inactive. This user has read access to objects in a few folders in the System cabinet of the repository only.

Content Server installation models

Content Server and repositories can be installed and configured in many different ways to meet various content management requirements. Content Server installation supports the following types of configurations:

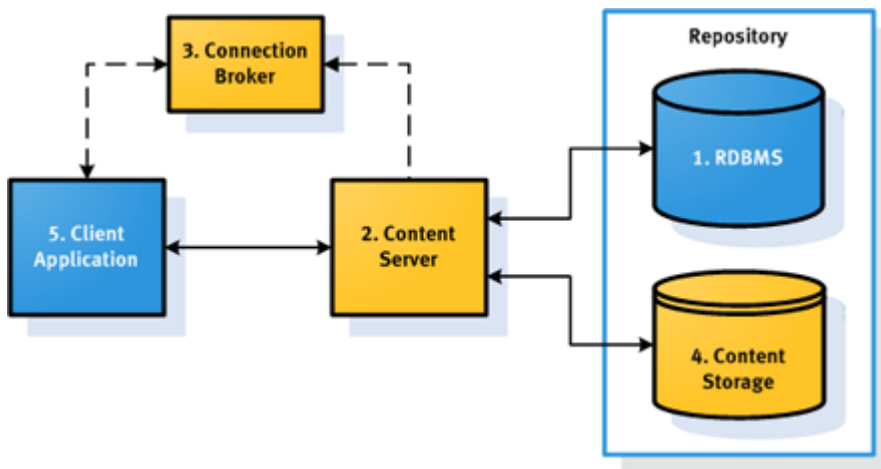
- Basic
In this basic model, Content Server, repository (including an RDBMS and a content storage), and connection broker are all installed on a single host. This is typically used in development and test environments.
- High-availability (HA)
In this model, multiple redundant Content Server instances and components are installed on a single host or multiple hosts and configured to eliminate single-point-of failure and achieve high-availability.
- Distributed
In the distributed model, one or more repositories span multiple hosts and are configured to be accessed from multiple sites.

This document focuses on the basic installation model to describe the Content Server installation process.

Basic installation model

Components that are installed as a part of the Content Server installation process covered in this document are highlighted in yellow. Dotted lines indicate connections that are not persistent.

Figure 2. Basic installation model



1. Relational database management system (RDBMS), which stores content metadata.

This is a part of the repository and a prerequisite software component that must be installed prior to Content Server installation.

2. Content Server, which manages content stored in the repository.
3. Connection broker, which provides connection information to client applications.
4. Content storage, which stores native content files.
5. Content Server client application, which provides a user interface for accessing Content Server functionalities and managing repositories.

Documentum Administrator is the primary web-based administrative client tool for configuring and administering Content Server and repositories.

Install the client application as a post-installation task.

Planning the installation

This section describes the installation decisions you should make before installing Content Server. These decisions will determine what prerequisite tasks you should perform, what choices to make and what information to provide during the installation process.

Use the Content Server installation checklist to plan your installation and record system information during installation and configuration. You can print out the checklist and keep it handy for reference.

Note: Information marked with an asterisk (*) next to it is prerequisite information you should obtain or plan for before installing Content Server.

Information	Note	Default Value
Content Server:		
Content Server host FQDN *	e.g. myhost.example.com	
Installation directory	<ul style="list-style-type: none"> Must not contain spaces On UNIX and Linux: no default; e.g. /usr/local/bin /Documentum 	C:\Documentum (Windows)
Installation owner username/password		
Application server administrator (admin) password		
Application server listening port *	A total of 20 ports, starting from the one you specify, will be used by the application server, and all of them must be available.	9080
Connection broker:		

Information	Note	Default Value
Connection broker connection mode	Native, Secure, or Native and Secure	
Connection broker name		
Connection broker port	The port you specify and its subsequent port will be used by the connection broker and both must be available.	1489
Connection broker host name *		
Repository:		
Data directory	<ul style="list-style-type: none"> Local, SAN, or NAS For SAN or NAS, enter the complete path including a shared device. For example: <code>\\x.x.x.x\folder1</code> <code>\folder2</code> where <i>x.x.x.x</i> is the IP address. 	<i>\$DOCUMENTUM/data</i>
Share directory		<i>\$DOCUMENTUM/share</i>
Repository name	<ul style="list-style-type: none"> Up to 32 alphanumeric characters beginning with a letter. The name <i>docu</i> reserved by the system. 	

Information	Note	Default Value
Repository ID	<ul style="list-style-type: none"> Any number from 1 to 16777215 and must not start with a zero (0). Must be unique on the network. The Repository ID is crucial for identifying all system related data, including configurations, objects, content, and so on and should be unique per repository. Each repository should have a unique Repository ID. These IDs are used to create file system paths in storage locations to ensure that they do not conflict with other repositories. However, if you create two repositories to have same Repository ID and point to same file location path, they might overwrite each others' data. This is because, for a particular storage path, the system assumes that it has full access to that path (which includes the Repository ID). 	
Authentication domain (Windows) *	The default domain if the user does not specify a Windows domain when connecting to the repository	
Repository connection mode *	Native, Secure, or Native and Secure	
ODBC data source name (DSN)		
Database administrator name/password *	The account has privileges to create and delete databases and perform other database administrative tasks	

Information	Note	Default Value
Repository owner name/password	The database user account with the database owner role that has read and write access rights to the database	
Repository database name	The name of the database Content Server uses to store content metadata as well as system and repository information	
Repository data device file path	The configuration program automatically fills in this information	
Repository log device file path	The configuration program automatically fills in this information	
SMTP server name *	SMTP server name or IP address that Content Server will use for email notifications	
Installation owner email address *	Installation owner's email address that Content Server will use for email notifications	
Global registry name	The central repository to manage installation-wide elements	
Global registry user login name		dm_bof_registry
Global registry user password		

Location of the content storage area (data directory)

Content Server installation creates the default content storage area called *the data directory* for storing content files. After the initial installation, you can add additional storage areas and the business rules for distributing content among them by using Documentum Administrator.

The data directory can be one of the following:

- A local directory on the Content Server host
- A directory on a remote host accessible by the Content Server
- A SAN or NAS location
- A location on other storage devices including retention type stores, such as EMC Centera and Network Appliance SnapLock

Prepare a location for the initial content storage area or the data directory that best suits your business needs. Ensure that the location you choose for the storage area has sufficient free space for the content files that will be added to it.

Ports to reserve for Content Server

Content Server and other components use a number of ports on the host:

- The application server listens on a port (9080 by default) for standard administration connections, and starting at this port, a total of 20 consecutive ports must be reserved for use by the application server.
- The connection broker requires two consecutive ports (1489 and 1490 by default) on which to listen: one for native connections and the other for secure connections (SSL).

On UNIX and Linux, the port numbers can be any unused port numbers greater than 1024. UNIX and Linux reserves port numbers up to 1024 for system use.

Identify available ports to be used by Content Server and its components. Make sure none of the reserved ports are being used for other purposes.

Connection modes to connect to connection broker and repository

There are three options you can choose from for the Content Server client to connect to the connection broker and the repository:

- **Native:** The client connects through a non-TLS/SSL port.
- **Secure:** The client connects through a secure TLS/SSL port. The client and the connection broker or repository do not use TLS/SSL authentication to authenticate each other. However, the information exchanged between them is encrypted.
- **Native and Secure:** The connection broker or repository accepts both native and secure connection requests.

Determine which connection mode you want to use for the connection broker and the repository respectively.

Content Server optional modules

Content Server has a number optional modules or extended features that require separate licenses. You activate a module by entering its license key in the Content Server configuration program. Once activated, you can enable the extended feature for a repository when you configure the repository.

Identify which optional modules to activate during Content Server installation and configuration and obtain license keys for those modules.



Caution: If you plan to use remote key management for your repository, you must activate the Trusted Content Services (TCS) module before you configure Content Server. If the TCS license is not present, you cannot create a repository that uses remote key management, and you cannot enable it at a later time.

Some of these optional modules, once activated, apply to all Content Server instances in an installation so that all the repositories can access the extended features; the others are individually

activated for Content Server instances, which means extended features can only be enabled for repositories managed by the Content Server instance with the corresponding modules activated.

The following modules are activated for Content Server instances in an installation across the board:

- High-Volume Server

High-Volume Server is an extension of Content Server that supports features implemented to solve common problems with large content stores. The three broad areas of enhancements that make up High-Volume Server are:

- Lightweight system objects
- Batch operations and currency scoping
- Database Partitioning

Note: Database partitioning option is not available while creating new repository during the Content Server installation. *EMC Documentum Content Server Administration and Configuration Guide* and *EMC Documentum High-Volume Server Development Guide* contains more information.

- Trusted Content Services

Trusted Content Services provides additional security features, such as encrypted file stores, in which content files are encrypted, and remote key management.

- XML Store

XML Store gives Content Server extended capabilities to store and process XML data in the repository by integrating it with Documentum xDB, a highly scalable, native XML database.

- Content Services for SnapLock

SnapLock is a feature of Filer, a NAS storage system from Network Appliance. A Network Appliance SnapLock (NetApp SnapLock) store stores large amounts of unchanging data such as email archives. SnapLock provides storage level retention capability through the creation of Write Once Read Many (WORM) volumes on Network Appliance storage systems. You can create a SnapLock volume in one of two modes: Enterprise and Compliance. The difference between the two modes is that on the Compliance SnapLock volume, a default (and minimum) retention of 30 years is applied to the content files created in that volume.

The following modules are activated for Content Server instances individually:

- Content Storage Services (CSS)

Content Storage Services enables you to define business rules to govern where content files are stored.

- Retention Policy Services (RPS)

Retention Policy Services is a compliance-oriented module that gives you control over how long and under what circumstances documents and other objects are retained in a repository.

Once you activate this module, you can use the event-based retention capabilities out of the box. However, if you want to use this module to manage retained data stored on a Centera device, you need another license to use the event-based retention capabilities and hold capabilities on that device.

- Federation Records Services

- Records Manager

Records Manager provides archiving options for business records.

EMC Documentum Content Server Fundamentals Guide and *EMC Documentum Content Server Administration and Configuration Guide* contains more information on modules and instructions on how to configure and use them.

Planning the system size

The system sizing spreadsheet is provided to assist with system planning and sizing. It focuses primarily on sizing deployments of the standard Documentum Editions. The spreadsheet automatically provide estimates of hardware resource needs based on user and hardware profiles provided. You can use this spreadsheet as an aid to size a Documentum deployment. This spreadsheet is available from the EMC Online Support within a few weeks after product release. You can download the EMC Documentum System Sizing Spreadsheet from the EMC Online Support site (<https://support.emc.com>).

Pre-installation requirements and tasks

Ensure that your system meets the system requirements and perform prerequisite tasks based on your installation plan before installing Content Server. You need the following:

- Administrative privileges on the computer where you are installing Content Server.
- Working knowledge of Microsoft Windows or UNIX and Linux, depending on which platform you choose for your Content Server installation.

Note: EMC recommends that you apply the latest operating system, .NET Framework, and related component patches. For example, operating system library updates, .NET Framework updates and so on.

System requirements

Consider the following in addition to the system requirements:

- RSA Data Protection Manager (Optional): If you want to create a repository to use remote key management, you must install RSA Data Protection Manager before installing Content Server.
- UNIX and Linux hosts: XWindows. XWindows must be installed on the UNIX and Linux hosts to run the graphical installation program. The xterm program may be installed in various locations depending on the operating system and software packages installed.

General database requirements

A relational database management system is a prerequisite to installing Content Server. The database stores metadata of content objects in the Content Server repository.

The database documentation contains information on installing and configuring the RDBMS.

Ensure that the RDBMS meets the following general requirements:

- Database location

You can install the relational database either locally on the Content Server host or remotely on a separate host running any operating system supported by the database vendor. For example, you can install Content Server on a Windows host and use a database installed on a UNIX and Linux host.

In a production environment, the database is almost always installed on a different host than the Content Server host for better performance.

- Database client

If you install the database on a separate host, also install the database client software on the Content Server host.

- For remote database installations, verify that you can connect to the database by using a database client from the system where you want to install Content Server.
- For local database installations on a UNIX and Linux host, verify that the system path includes the installation directory for the database.

On a Windows host, the installer updates the system path automatically.

- Database code page

For new repositories, install the database by using the Unicode code page, which can accurately store characters from all supported languages.

- For new SQL Server repositories, Content Server uses only `nvarchar` and `nchar` types, which automatically use Unicode. If you need to determine the settings of an existing SQL Server database, use the system stored procedure `sp_helpsort` or view the properties of the particular database in Enterprise Manager.
- On Oracle and DB2, use UTF8 (this includes AL32UTF8 and AL16UTF8).

Note: Oracle has deprecated AL16UTF16. AL16UTF16 is supported for now, but it is recommended to migrate to AL32UTF8 or AL16UTF8 code page.

- Database locales

Typically, Content Server is installed on the English version of database. However, Content Server installation is also supported on localized databases if the database fulfills the following criteria:

- Database supports internationalization of locales (I18N)
- Database adheres to I18N standards
- Content Server installation is performed with UTF8 and case sensitive (SQL)

- Database administrator account

Regardless of the database you use with Content Server, obtain the database administrator's username and password. You might need this information while configuring the repository.

- Database service (Windows)

If Content Server and the database are located on the same Windows host, ensure that the database service is set to start automatically. Content Server installation sometimes requires a restart of the computer. After the restart, installation does not proceed correctly unless the database starts automatically.

Requirements for Microsoft SQL Server

If you use Microsoft SQL server with Content Server, ensure it meets the following requirements:

- Use a full SQL Server installation on the host where SQL Server is installed. Install the SQL Server Management Studio and SQL Server client on the Content Server host, regardless of whether the database is local or remote. You need to install the required client packages from SQL Server installation for Content Server to work remotely with the SQL Server database.
- Use the Custom installation option so that you can set the database code page, case-sensitivity, and other options.
- Install the SQL Server instance in SQL Server or Windows Authentication mode.
- Install SQL Server for internationalization. The SQL Server documentation and Microsoft MSDN website contains more details.
- Ensure that the SQL Server sort order is set to **Dictionary**.
- Ensure that you configure the SQL Server with the following settings:
 - PARAMETRIZATION=FORCED
 - Max Degree of Parallelism = 1
 - Allow Snapshot Isolation = ON
 - READ_COMMITTED_SNAPSHOT = ON

Ensure that the database statistics are up to date and optimal. Configure the database based on the results after profiling the database usage.



Caution: Install the database in case-sensitive mode with row-level locking enabled. If you installed SQL Server in case-insensitive mode, you need to reconfigure the database before you install or upgrade Content Server.

Requirements for Oracle database

If you use Oracle database with Content Server, ensure it meets the following requirements:

- If Content Server resides on a host different from the Oracle database, you need to install the Oracle Database Client package on the Content Server host.
- On UNIX and Linux, ensure that these environment variables are set in the installation owner's environment:
 - ORACLE_HOME
 - TNS_ADMIN
 - ORACLE_SID

This environment variable points to the location of the `tnsnames.ora` file. Content Server installer looks first for `TNS_ADMIN`, then for `ORACLE_HOME`, in order to locate the `tnsnames.ora` file.

- For Windows, UNIX, and Linux hosts, install the 64-bit Oracle Client package on the Content Server host and update the value of the shared library path environment variable (LIBPATH for AIX or LD_LIBRARY_PATH for Linux and Solaris) to include the library directory.
- Ensure that you configure the following settings:
 - `Cursor_sharing=Force`
 - Automatic memory management is enabled.
- Increase the size of the REDO log files from the default value of 50 MB and/or increase the number of REDO log files if the database runs into heavy updates.
- Ensure that the database statistics are up to date and optimal. Configure the database based on the results after profiling the database usage and by monitoring database reports such as AWR report, Oracle Alert log, Query execution plans, and so on. The following parameters can also impact the performance:
 - `Session_cached_cursors`
 - `Processes`
 - `MEMORY_TARGET`
 - `MEMORY_MAX_TARGET`
 - `Pga_aggregate_target`
 - `SGA_TARGET`
 - `SGA_MAX_SIZE`
- Environment variable `DM_CHECK_EMPTY_STRING_IN_ORACLE` handles the '' (empty strings) in queries. If `DM_CHECK_EMPTY_STRING_IN_ORACLE` is not set (by default) or if `DM_CHECK_EMPTY_STRING_IN_ORACLE = 1` then '' is treated as '' so that query results is same for all databases.
- Environment variable `DM_DEGREE_OF_PARALLELISM_ORACLE` is introduced to enable the parallel indexing on Oracle database. The value of the variable must be set to an integer (for example, 4) which is used as Degree of Parallelism (DOP) while creating indexes in Oracle database.

Requirements for DB2 database

The DB2 configuration requirements apply if DB2 and Content Server are running on AIX.

If you use DB2 database with Content Server, ensure it meets the following requirements:

- On AIX, ensure that the following DB2 environment variables are set in the installation owner's environment:
 - `DB2_BASE`
This must point to `/DB2_installation_dir/home/instance_name/sqlib`.
 - `DB2INSTANCE`

This must point to the name of the default DB2 instance.

Ensure that the LIBPATH environment variable includes \$DB2_BASE/lib.

- To support audit trail functionality, DB2 requires 8K-page capability. During the installation of version 6.5, the installer automatically creates 8K pages. To find out whether you have 8K temporary tablespace before an installation or upgrade, run the following command:

```
db2 LIST TABLESPACES SHOW DETAIL
```

If the page size parameter is 4096, you have 4K page size, if it is 8192, you have 8K page size.

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

```
db2 CREATE TEMPORARY TABLESPACE TEMPSPACE2 PAGESIZE = 8192
```

- Before you create a database for use by Content Server, disable the DB2CODEPAGE environment variable from the command line:

```
db2set DB2CODEPAGE =
```

After you create the database, start the DB2 command line. From the command line, set the DB2CODEPAGE environment variable to 1208:

```
db2set DB2CODEPAGE=1208
```

- Ensure that the DB2 clients are installed on the Content Server host.
 - If you install DB2 on the same host as the Content Server, the clients are installed automatically.
 - If you install DB2 on a different host from the Content Server, you need to manually install the DB2 clients on the Content Server host.
- Set the code page to UTF-8.

Note: Do *not* set the environment variable DB2OPTIONS. If set to T, the DB2 command-line processor uses a semicolon (;) as the statement termination character. Content Server does not install properly on AIX with DB2 when DB2OPTIONS is set.

Requirements for DB2 performance wizard

Using the DB2 performance wizard has the following requirements:

- Set the server memory target value.
 - If DB2 is installed on the Content Server host and you are installing DB2 Enterprise Edition, set the target memory to 40%.
 - If DB2 is installed on a different host from Content Server and you are installing DB2 Enterprise Edition, set the target memory to 80%.
- Ensure that the buffpage value is at least 6000.

Requirements for PostgreSQL database

If you use PostgreSQL database with Content Server, ensure it meets the following requirements:

- Install the PostgreSQL server and client and configure the `iptables`.
- Configure `ODBC.INI` as follows:

```
[MyPostgres]
Description = PostgreSQL connection to MyPostgres
Driver = PostgreSQL
Database = postgres
Servername = Server where PostgreSQL is running
Username = postgres
Password = user password
Port = 5432
Protocol = 7.4
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
UpdateableCursors = Yes
DEBUG = Yes
```

- Configure ODBCINST.INI as follows:

```
[PostgreSQL]
Description = ODBC for PostgreSQL
Driver = /usr/pgsql-9.4/lib/psqlodbcw.so
Driver64 = /usr/pgsql-9.4/lib/psqlodbcw.so
Setup64 = /usr/lib64/libodbcpsqlS.so
FileUsage = 1
```

- Log in as a postgres user and create a directory called db_reponame_dat.dat in /var/lib/pgsql/9.4/data/.
- On Windows:
 - Change the value of Protocol in ODBC.INI to **7.4-2**.
 - Perform the following configuration while creating an ODBC data source:
 - Create a driver with the UNICODE option.
 - Disable the LF<->CR/LF conversion option.
 - Enable the **Updatable Cursors** option.
 - Navigate to **System DSN > Datasource** and remove **dd_** from **SysTable Prefixes**.
- On UNIX, Linux, CentOS, and Ubuntu:
 - Replace the psqlodbcw.so and psqlodbc.so library files. PostgreSQL client library must be a non-stripped version. For example:

```
[root@postgres bin]# file /usr/pgsql-9.2/lib/psqlodbcw.so
/usr/pgsql-9.2/lib/psqlodbcw.so: ELF 64-bit LSB shared object, x86-64,
version 1 (SYSV), dynamically linked, not stripped
```
 - Change the value of Servername in ODBC.INI to **localhost**.

- To allow remote connection:
 - Change the value of the method parameter to **trust** instead of **ident** in `/var/lib/pgsql/data/pg_hba.conf`. For example:
 - Use local for UNIX domain socket connections only: `local all all trust`
 - IPv4 local connections: `host all all 127.0.0.1/32 trust`
 - IPv6 local connections: `host all all ::1/128 trust`
 - Change the value of `listen_address` to `'*'` in `/var/lib/pgsql/9.4/data/postgresql.conf`.
- By default, PostgreSQL does not enable the clustered index. Hence, perform the following:
 - Create a clustered index for `r_object_id` column for all tables.
 - Force the clustering for all the clustered tables. To do this, in `CreateIndex()` method (in `dbxx` layer), execute the following DDLs:

```
alter table <table_name> CLUSTER ON <index_name>. CLUSTER <table_name>
```

When the existing table is subsequently updated, the changes are not clustered. No attempt is made to store new or updated rows according to their index order. So, recluster by issuing the `CLUSTER` command again. To achieve this, expose a new parameter, `DM_CLUSTER_INDEX` to the `POST_UPGRADE_ACTION` apply method. When this is called with the `table_name`, force clustering is applied to that table. If `table_name` is not specified, force clustering is applied to all tables.

Use the following apply method:

```
apply,c,NULL,POST_UPGRADE_ACTION,EXECUTION_MODE,S,
DM_CLUSTER_INDEX,TABLE_NAME,S,dm_location_s,TRACE_ON,B,T,
FORCE_FLAG,B,F
```

- To update the `i_partition` value of any object to another partitioned table range on a data partition enabled CentOS/PostgreSQL environment, enable the `constraint_exclusion` parameter. Since PostgreSQL uses the concept of constraint exclusion to enable partition boundary checking, you must set the `constraint_exclusion` parameter in the `postgresql.conf` file. Open the `postgresql.conf` file, navigate to the `QUERY TUNING` section and remove the pound sign (`#`) for the `constraint_exclusion` entry. Set the `constraint_exclusion` parameter to **on**, and save the `postgresql.conf` file before exiting.

Note:

- Batching, specifically the delayed commit on batches, is not supported in PostgreSQL.
- For sorting issues with the `ORDER BY` clause, ensure that you add `r_object_id` in the select statement so that the Content Server explicitly adds `ORDER BY r_object_id` in the select statement.

Enabling data partition in PostgreSQL database

1. Run the following command in DQL:

```
EXECUTE partition_operation WITH "operation"='create_scheme',
"partition_scheme"='<name of the scheme>',
"partition_name"='<name of the first partition, for example P1>',
"range"='<specify the range, for example 100>',
"tablespace"='DM <repository name> _DOCBASE',
"partition_name"='<name of the second partition>',"range"='<specify the range>,'
```

```
"tablespace"='DM_<repository name>_DOCBASE',  
"partition_name"='<name of the nth partition>',"range"=<specify the range>,  
"tablespace"='DM_<repository name>_DOCBASE'
```

2. Run the following command in DQL:

```
EXECUTE partition_operation WITH "operation"='db_partition',  
"partition_scheme"='<name of the scheme>'
```

3. Run the following command in DQL:

```
EXECUTE get_file_url FOR  
<object id generated from Step 2> WITH "format"='text'
```

4. Run the following command in IAPI:

```
getpath,<session id>,  
<object id generated from Step 2>  
This generates the database script to enable the data partition.
```

5. Stop the Content Server.
6. Connect to the PostgreSQL database as a repository owner and run the script generated from [Step 4](#).
7. After enabling the data partition, run the following database queries:

```
drop view dm_resync_dd_attr_info;  
drop view dm_resync_dd_type_info;  
drop view dm_dd_policies_for_attrs;  
drop view dm_dd_policies_for_type;  
delete from dmi_vstamp_s where i_application='dm_dd_attr_info_view_tag' or  
i_application='dm_dd_type_info_view_tag' or  
i_application='dm_dd_policies_views_stamp';
```

8. Start the Content Server.

Note:

- All the supertype partitions are inherited to the current type. If none of the supertypes are partitioned, then you have to enable data partition for the corresponding type.

For example:

```
create type <type name> (<name of the attribute 1> <type>,  
<name of the attribute n> <type>) with supertype dm_sysobject
```

The created type inherits all the partitioned properties from dm_sysobject.

- Creating a partitionable type is not supported. As an alternative, you can create a custom type object and enable data partition only to this custom type using EXECUTE partition_operation with db_partition by specifying the value for the type_name attribute. For example:

```
create type <type name, for example 'example_type'>  
(<name of the attribute 1> <type>, <name of the attribute n>  
<type>) with supertype NULL  
  
EXECUTE partition_operation WITH "operation"='db_partition',  
"partition_scheme"='<scheme name>',"type_name"='example_type'
```

Continue from [Step 3](#) of the [Enabling data partition in PostgreSQL database, page 95](#) procedure to create a partitionable type.

Also, you may encounter missing i_partition attribute for custom type objects after partitioning. To fix, after the data partitioning script is run against the database server successfully, restart the

Content Server and then run the following DQL statement to complete the type partitioning process:

```
ALTER TYPE <type_name> ENABLE PARTITION
```

Tuning PostgreSQL database

1. Append the following lines at the end of <path>\<to>\postgresql.conf:

For example (PostgreSQL 9.4):

```
temp_buffers = 32MB
work_mem = 32MB
checkpoint_segments = 32
checkpoint_timeout = 10min
checkpoint_completion_target = 0.5
random_page_cost = 2.0
default_statistics_target = 500
maintenance_work_mem = 256MB
shared_buffers = <1/4 times of physical memory>
effective_cache_size = <3/4 times of physical memory>
wal_buffers = 16MB
synchronous_commit = off
```

2. Restart the PostgreSQL instance.
3. Login on the psql command line prompt and check for modified parameter to verify if the changes are applied as:

```
# show random_page_cost;
```

4. Create new pattern indexes:

- dm_sysobject_s:

```
create index test_ops_idx on dm_sysobject_s (r_object_id bpchar_pattern_ops);
```

- dm_sysobject_r

- dm_acl_s:

```
create index test_dm_acl_s_idx on dm_acl_s (r_object_id bpchar_pattern_ops);
```

5. Execute *cluster* for clustered indexes maintenance:

```
# cluster;
```

6. Recreate all indexes in the database to remove index fragmentation:

```
# reindex database "dm_DOCREPO_docbase";
```

7. Execute *vacuum full* and analyze to remove table fragmentation and also update database statistics:

```
# VACUUM FULL VERBOSE ANALYZE;
```

8. Run the query to check dead tuples in the database:

```
SELECT psut.relname,
to_char(psut.last_vacuum, 'YYYY-MM-DD HH24:MI') as last_vacuum,
to_char(psut.last_autovacuum, 'YYYY-MM-DD HH24:MI') as last_autovacuum,
to_char(pg_class.reltuples, '9G999G999G999') AS n_tup,
to_char(psut.n_dead_tup, '9G999G999G999') AS dead_tup,
to_char(CAST(current_setting('autovacuum_vacuum_threshold') AS bigint)
+ (CAST(current_setting('autovacuum_vacuum_scale_factor') AS numeric)
* pg_class.reltuples), '9G999G999G999') AS av_threshold,CASE
```

```
WHEN CAST(current_setting('autovacuum_vacuum_threshold') AS bigint)
+ (CAST(current_setting('autovacuum_vacuum_scale_factor') AS numeric)
* pg_class.reltuples) < psut.n_dead_tup
THEN '*'
ELSE ''
END AS expect_av
FROM pg_stat_user_tables psut
JOIN pg_class on psut.relid = pg_class.oid
ORDER BY 1;
```

You should not see a huge number for the `dead_tup` column.

9. Run the query to check index fragmentation:

```
with indexBloat as
(
SELECT nspname as schema, c.relname as table_name,
i.relname as index_name,
ROUND(ROUND(100 * pg_relation_size(indexrelid) / pg_relation_size(indrelid), 2)
/ 100, 2) AS iratio,
pg_size_pretty(pg_relation_size(indexrelid)) as index_size,
pg_size_pretty(pg_relation_size(indrelid)) AS table_size,
pg_relation_size(indexrelid) as isize_byte
FROM pg_index x
JOIN pg_class c ON c.oid = x.indrelid
JOIN pg_class i ON i.oid = x.indexrelid
JOIN pg_namespace n ON (n.oid = c.relnamespace)
WHERE nspname NOT IN ('pg_catalog', 'information_schema', 'pg_toast')
AND i.relkind = 'i'
AND c.relkind = 'r'
AND pg_relation_size(indrelid) > 0
)
SELECT *
FROM indexBloat WHERE schema != 'snapshots'
ORDER BY isize_byte desc;
```

For any index (> 50MB), the `iratio` should be less than 80%. Only if the index is composite, the `iratio` should be close to 80-85% and not higher than the table itself. Ignore empty or small tables (8KB or few KB tables).

Note: PostgreSQL performance gap compared to the Linux/Oracle and Windows/SQL Server combinations is about 13% to 14% and 16% to 19% respectively for 100 and 400 users.

Setting up required user accounts

Before installing Content Server, set up required user accounts in the host operating system and RDBMS.

Installation owner account

The *installation owner account* is an operating system account with appropriate permissions to install Content Server and create repositories. The installation owner account may be a local account on the Content Server host or a domain account in the domain where Content Server is installed. The account must be a member of the Administrators group on the local host.

Content Server runs under the installation owner account. The installation owner can perform all administrative and maintenance tasks associated with Content Server and repositories.

For security reasons:

- On Windows, it is recommended that the installation owner account is not the same account as the Windows Administrator.
- On UNIX and Linux, it is recommended not to use the root account as the installation owner account.

You can create an operating system account to use exclusively for Content Server installation and repository configuration. You can use a single operating system account as installation owner for multiple Content Server installations on the network.

If you use Windows authentication for SQL Server, the Content Server installation owner must have the System Administrator privileges in SQL Server.

Note:

- After installation, the installation owner account cannot be changed.
- The database user specified in the `server.ini` file as `database_owner` should have an account with the database.
- The installation owner account should have full access to `AEK.KEY` and `DBPASSWD.TXT`.
- In a multi-user account AIX system, resources created or owned by a user account may not be available for modification to another user account due to access restrictions. This might lead to error such as `DM_CRYPTO_E_SHM_CREATE_FAILED` if the shared memory is locked by another user account. Ensure that a proper user account planning is done to avoid these conflicts. Also, sometimes a system reboot may be necessary to clean up all the resources of the process such as shared memory.

Installation owner account naming requirements

The installation owner username must consist of letters, numbers, dashes (-) or underscores (_). The first character must be a letter. All characters must be ASCII characters.

The installation owner password consist of letters, numbers, dashes, underscores, or periods.

Required rights for the installation owner account

The installation owner account must have the following user rights on the host operating system:

- Act as part of the operating system
- Create a token object
- Increase quotas
- Log in as a service
- Log in locally
- Replace a process-level token

On Windows, these rights are automatically inherited with membership in the local Administrators group. Installer checks for these rights and grants them if necessary.

The installation owner must have Full Control permission on the directories into which Content Server is being installed, including data and share directories. The installation owner must also have write permission on the directory from which installer is run.

On UNIX and Linux, the installation owner must have read, write, and execute permission on the */tmp* directory.

To support external password validation, set up a group account whose members are the installation owner, any other Content Server administrators, and repository owners. This will be the group that owns the external password validation program.

Installation owner's email account and SMTP server information

Content Server requires the installation owner's email address and SMTP server information to send email notifications to the installation owner. During installation or upgrade, you need to provide this information and installer will attempt to connect to the SMTP server. The SMTP server can be located on the Content Server host or another computer on the network.

Set up an email account for the installation owner and obtain the host name or IP address of the computer hosting the SMTP server Content Server can connect to.

Repository owner account

The *repository owner account* is a database user account that Content Server uses to connect to the database. This account owns all objects in the database and gives Content Server access to the database tables underlying the repository. Each repository must have a unique repository owner.

You can create the repository owner (database user) account in one of these two ways:

- Allow Content Server to automatically create the repository owner account in the database when you create the repository during the installation process.

The configuration program automatically grants the account proper privileges.

- Manually create the repository owner account in the database prior to installing Content Server.

Ensure that the account has appropriate privileges to perform the following tasks:

- Connect to the database
- Create tables, views, and indexes in the database

- Insert records (rows) into the tables
- Drop tables, views, and indexes

Note the following requirements for different RDBMS:

- Microsoft SQL Server
 - The repository owner must be able to access tempdb, and if the account is created before installer is run, the user must own all tables and views. Ensure that the repository owner has the Create Any Database privilege.
 - On Windows, if you use Windows authentication for SQL Server, the repository owner (database user) must have a Windows account.
- Oracle database
 - If you create the account before Content Server installation, provide a value for the `select_catalog_role` parameter.
 - The repository owner must have Connect and Resource privileges. The Resource privilege encompasses creating and maintaining database objects. The repository owner also must have permission to create any view, resource, and unlimited tablespace. The tablespace created by the repository owner for tables or indexes must be designated the default, while the standard Oracle temporary tablespace must be the default for any temporary tables that the repository owner creates. The name of the temporary tablespace must be valid for the Oracle configuration used. The default name is either `temporary_data` or `temp`, depending on the version of Oracle.

The repository must also have the Select Catalog Role privilege.
- DB2 database
 - Grant use of tablespaces, list tablespace, and connect to database privileges.
 - The repository owner does not have an account. The repository owner is created when you grant the required privileges to an existing operating system account.

To use Microsoft Cluster Services, the repository owner must have an account in the domain in which you install the repository.

Repository user accounts

Repository users are the end users in the repository.

- On Windows, if the default user authentication is used, each user must have a Windows account in the domain where Content Server is installed. If LDAP authentication or inline password authentication is used, this is not a requirement.
- On UNIX and Linux, if the default user authentication is used, each user must have an operating system account in the domain where Content Server is installed. If LDAP authentication or inline password authentication is used, this is not a requirement.

Pre-installation tasks on Windows

Perform the following pre-installation tasks on Windows:

- Set the date and time formats to a four-digit year (yyyy) date in the Windows regional settings.
- Disable the user access control (UAC).
- Disable the Windows Update service.
- Enable the Computer Browser service (optional).
- Disable the IP Helper service from the Windows Services console and restart the machine. This method disables the Teredo Tunneling Pseudo-Interface.
- In non-English operating systems, install the latest version of Microsoft Visual C++ 2008 Redistributable (64-bit) before creating a repository. This will provide the correct operating system runtime libraries for the Content Server and other utilities.
- Install the Microsoft security updates released in June, 2014 onwards to avoid vulnerabilities on the Windows hosts.

Pre-installation tasks on UNIX and Linux

Performing pre-installation tasks for SUSE Enterprise Linux

Before installing the Content Server, ensure that you have done the following configuration:

- To ensure that the `dmdb` test does not fail during the loading of the shared libraries of `libssl.so.10`, while configuring the repository, perform the following steps:

```
root:~ # cd /lib64
root:/lib64 # ln -s libcrypto.so.1.0.0 libcrypto.so.10
root:/ # cd /usr/lib64
root:/usr/lib64 # ln -s libssl2.so.3 libssl2.so.2
root:/usr/lib64 # ln -s libssl3.so libssl.so.10
```
- The repository might crash while loading the Netegrity plugin. Hence, ensure that you copy `libstdc++.so.5` from the older version of SUSE and save it at `/usr/lib/` in SUSE Enterprise Linux 12 environment.

Performing pre-installation tasks for Red Hat Enterprise Linux 7.x

Before installing the Content Server, ensure that you have done the following configuration:

- To ensure that the repository configuration does not fail with `libsasl2.so.2: cannot open shared object file error` on Red Hat Enterprise Linux 7.x, log into the `root` account and run the following command:

```
ln -s /usr/lib64/libsasl2.so.3.0.0 /usr/lib64/libsasl2.so.2
```

Setting the required environment variables

In the installation owner's environment, set environment variables that identify the directories into which Content Server will be installed. The variables must be set in the installation owner's environment.

You can create the Content Server installation directory before installing the server, or you can allow the installer to create the directory for you during installation. If you allow the installer to create the directory, ensure that the directory name you provide during the installation matches the one specified in the environment variable.

Symbolic links (symlinks) are not supported. Do not use them in the installation directory or any other environment variable used by Content Server. This restriction also applies to environment variables used to specify the database location. For example, if you use Oracle, the ORACLE_HOME environment variable cannot use a symbolic link. However, the symbolic links for Java upgrade is supported.

Environment variables and the installation directory must contain only ASCII characters. The name of the installation directory must not contain spaces.

On UNIX and Linux, you need to set certain environment variables in the installation owner's environment.

You must set the `$DOCUMENTUM` variable first for installer to run successfully.

- `$DOCUMENTUM`

The Content Server installation directory. The installation owner must have read, write, and execute permission on the `$DOCUMENTUM` directory and its subdirectories. There is no default installation directory on UNIX and Linux, and `$DOCUMENTUM` cannot be mounted with the `nosuid` option.

The Content Server configuration program (`dm_launch_server_config_program.sh`) automatically sets all required environment variables except for those required by each database. If you do not use the Content Server configuration program, you need to manually set all environment variables.

You can set all of the following variables, except `LC_ALL` and `DISPLAY`, by sourcing `$DOCUMENTUM/product/version_number/bin/dm_set_server_env.sh` or `$DOCUMENTUM/product/version_number/bin/dm_set_server_env.csh`. Set the variables `LC_ALL` and `DISPLAY` in the installation owner's `.cshrc` file (C shell) or `.profile` file (Bourne or Korn shells). Alternatively, set the variables in a file called by the `.cshrc` file or `.profile` file or in other ways permitted by UNIX and Linux.

- `$DM_HOME`

Must be `$DOCUMENTUM/product/version_number`, where `version_number` is the version of Content Server.

- `LIBPATH` (AIX) and `LD_LIBRARY_PATH` (Linux and Solaris)

Add the directory containing the client library to the appropriate environment variable of the installation owner. You must set this environment variable before configuring or starting Content Server, as well as running IAPI or IDQL.

- DISPLAY

Controls the display. Set the value to `<host/ip>:0.0`.

- LC_ALL

Set the value to C.



Caution: If this value is not set correctly, the Java Method Server will fail.

Note:

- On AIX, depending on the environment and the customizations, Content Server could exceed the default process memory size limit that is set by the AIX operating system. This default in 64-bit process is 256 MB on AIX. While configuring Content Server, do some analysis on the number of sessions, users, operations and set the LDR_CNTRL environment variable appropriately. The recommended value is 2 GB which is LDR_CNTRL=MAXDATA=0x7000000.

If the load of the Content Server is not known in advance and the LDR_CNTRL environment variable is not set properly, Content Server could throw a Segmentation Violation error with message "Error: dm_bear_trap: Unexpected exception, (SIGSEGV: segmentation violation: (11) at (Connection Failure))".

When this error occurs, modify the repository startup script `$DOCUMENTUM/dba/dm_start_<docbase>.sh` and add the below setting before launching the Content Server and then restart the Content Server:

For example:

```
LDR_CNTRL=MAXDATA=0x7000000
export LDR_CNTRL
./documentum -docbase_name <docbaseName>
-security acl -init_file /path/to/server.ini $@ >> $logfile 2>&1 &
```

- On a AIX with DB2 combination, depending on the number of processes that need to be run, you should change the user process.

Command to check the user process:

```
lsattr -E -l sys0 -a maxuproc
```

Command to increase the user process:

```
chdev -l sys0 -a maxuproc= 256 (or more)
```

- On a AIX with DB2 combination, when you install or upgrade Content Server 7.1 and later, set the file descriptor limit to any 5 digit value (for example, **65535**) and do not use **unlimited**.

Setting up the services file

The services file must contain two entries for each repository running on a host. Manually create the service name entries in the services file before you install the Content Server.

The repository does not have default service names or default port numbers. The service name you put in the services file must be the same name you will provide during repository configuration, which is then used to create the `server.ini` file. The service name for the repository can be the same as the repository name, but this is not required.

The services file must include entries that designate two consecutive port numbers: one for native connections, and the other for secure (SSL) connections. Append `_s` to the name of the repository service for the secure connections.

Create two network service entries in the system's service table using the following format:

```
service_name port_number/tcp #Put comments here
service_name_s port_number/tcp #Put comments here
```

In the following repository service entries example, repository service is named `rep01`.

```
dm_rep01
47625/tcp # <version_number> Repository native connection
dm_rep01_s
47626/tcp # <version_number> Repository secure connection
```

Note:

- Even if you are not using secure (SSL) connections, two consecutive port numbers are still required by Content Server.
- If correct services file entries are not present during installation, the installation will fail.

If you have multiple repositories on a single host, create a services file entry for each repository. Ensure that the repositories use different names and port numbers.

Installing the JCE policy files

To use all ciphers on a Oracle Java environment, download the Unrestricted SDK JCE policy files from the Oracle website. The downloaded files include the `local_policy.jar` and `US_export_policy.jar` policy files. Take a copy of existing policy files from the Java home directory and save it in another directory. Copy the downloaded policy files in the `<Java_Home>/jre/lib/security` directory for both the Content Server and the client machines.

Note: To use all ciphers on a AIX with Oracle combination, download the Unrestricted SDK JCE policy files from the IBM website.

Enabling random generator on Linux

A connection request through SSL waits for a random number to be created and a delay is noticed. To avoid the delay, you can enable or disable the random generator on the Linux machine. Perform the following boot safe (inittab) on all Content Server machines:

Start the random generator as root: `/sbin/rngd -b -r /dev/urandom -o /dev/random`

Preparing relational database management system (RDBMS)

Creating an ODBC data source for Microsoft SQL Server

On the Content Server host, create an ODBC data source for SQL Server. Please note the following:

- Ensure that you select a 64-bit ODBC data source using only the SQL Server driver.
- You can also use **User DSN** tab.
- Options when you choose a SQL Server authentication method:
 - **With Windows NT authentication using the network login ID:** If you choose this option, the Content Server installation owner must have the System Administrator privileges in SQL Server and the repository owner (database user) must have a Windows account.
 - **With SQL Server authentication using a login ID and password entered by the user:** If you choose this option, enter the SQL Server administrator login ID and password. The repository owner (database user) does not need to have a Windows account.

Configuring Oracle database

Configuring the tnsnames.ora file

Oracle database aliases (TNS aliases) are defined by entries in the `tnsnames.ora` file. Configure the `tnsnames.ora` file on the Content Server host. Use the Oracle SQL*Net configuration tool to create a database alias referring to the database instance you plan to use for Content Server. After you create the alias, test the alias by connecting to the Oracle database.

Entries in the `tnsnames.ora` file for the Oracle HTTP service and data expo service do not contain parameters for HOST, SID, and SERVICE. If the first entry in the `tnsnames.ora` file is for one of these services, Content Server installer is unable to parse the `tnsnames.ora` file and cannot connect to the database. Ensure that the first entry in the `tnsnames.ora` file is not for the Oracle HTTP service or data expo service.

To install Content Server with Oracle RAC, update the Oracle machine name or IP address in `tnsnames.ora` with SCAN name or SCAN IP address. Ensure that you maintain the same Oracle database Service_Name when you migrate non-RAC Oracle database to Oracle RAC database.

Note:

- Content Server supports Oracle RAC only as a standalone support. Content Server does not support failover/load balancing features of Oracle RAC.
- Content Server provides limited support for Oracle RAC. Content Server supports only those features that are natively supported per the Content Server design.

The `database_conn` key in the `server.ini` file must match the database entry in the `tnsnames.ora` file. If it does not, an error occurs. To fix, modify the `database_conn` key in the `server.ini` file and continue with the installation or upgrade.

Performing additional Oracle database configuration tasks

Perform the following Oracle database configuration tasks before installing Content Server:

- Set up networking parameters.
- Ensure that the Oracle listener is running on the Oracle host.
- Optionally, migrate an existing Oracle instance to AL32UTF8 or AL16UTF8 code page.
- Perform the following to support Oracle multi-tenant container database:
 - Create a net service for the PDB service in `tnsnames.ora`.
 - Set `USE_SID_AS_SERVICE_LISTENER = ON` in `listener.ora` to allow using service name as SID.
- When PDB is configured, it will be in mounted state. You must provide the read write permission.

Configuring DB2 database

You can configure a DB2 database by using one of the following methods:

- Configure DB2 from the Control Center
- Configure DB2 from the command line

Optionally, you can use the performance wizard to fine-tune DB2 performance after installing and configuring the database. You can use the performance wizard at a later time, but if you do so, ensure that the parameter values required by Content Server are not changed.

General guidelines

You can create the database and set the parameters from the DB2 command line or from the Control Center. Use the Control Center to run the performance wizard. You can run the Control Center on AIX, or you can run performance wizard from a Windows system to tune performance for the instance on AIX. Install and configure DB2. You can use the performance wizard at a later time (after you complete configuring DB2), but if you do so, ensure that the parameter values required by Content Server are not changed.

Configuring DB2 from Control Center

Use the following procedure to configure a DB2 database from the Control Center:

1. Start the Control Center.
2. Right-click the database and choose **Configure** from the context menu.
3. Click the **Performance** tab.
 - a. Set the sort heap.

If you are configuring DB2 Enterprise Edition, set the sort heap to 1138.

- b. Set the application heap size to 1024.
 - c. Set the application control heap size to 256.
 - d. Set the log buffer size.
If you are configuring DB2 Enterprise Edition, set the log buffer size to 128.
 - e. Set the lock list size.
If you are configuring DB2 Enterprise Edition, set the lock list size to a minimum of 2048.
4. Click the **Applications** tab and set the maximum number of locks to 80.
5. Click the **Logs** tab.
 - a. Set the log file size to 1024.
 - b. Set the number of primary logs to 18.
 - c. Set the log second size.
If you are configuring DB2 Enterprise Edition, set the size to 10.Click **OK** and close the dialog box.
6. When the Control Center is displayed, click the database for your repository and right-click **Buffer Pools**.
7. Choose **Alter**. The Alter Buffer Pool dialog box displays.
8. Check **Use default bufferpool size**, and click **OK**.
9. From the command line, restart the DB2 server:

```
db2stop force
db2start
```

Configuring DB2 database from command line

Use the following procedure to configure a DB2 database from the command line:

1. Start the DB2 command line.
2. Set the application heap size to 1024 or greater, where <dbname> is the name of the database you created for use by the repository:

```
update db cfg for <dbname> using applheapsz 1024
```
3. Set the application control heap size to 256 or greater

```
updatedb cfg for <dbname> using APP_CTL_HEAP_SZ 256
```
4. Set the transaction file sizes.
If you are configuring DB2 Enterprise Edition:

```
update db cfg for <dbname> using LOGFILSIZ 1024
update db cfg for <dbname> using LOGPRIMARY 18
update db cfg for <dbname> using logbufsz 128
update db cfg for <dbname> using logsecond 10
```
5. Set the maximum number of locks:

```
update db cfg for <dbname> using maxlocks 80
```

6. Set the lock list size.

On the DB2 Enterprise Edition, set it to a minimum of 2048:

```
update db cfg for <dbname> using locklist 2048
```

7. Set the sort heap and buffer page sizes. Ensure that the buffer page size (buffpagesize) is set to a minimum of 6000.

If you are configuring DB2 Enterprise Edition:

```
update db cfg for <dbname> using sortheap 1138
update db cfg for <dbname> using buffpage buffpagesize
```

If the repository and the DB2 server are on the same machine and you are configuring DB2 Enterprise Edition, set buffpagesize to 40% of the available physical memory divided by the page size of your tablespace.

If the repository and the DB2 server are on different machines, set buffpagesize to 80% of the available physical memory divided by the page size of your tablespace.

Note: If you are running more than one repository and database on the same DB2 server, the percentage recommended for buffer pool is for the sum of all databases. In all cases, ensure that the buffer page size is a minimum of 6000.

8. If you see the SQL1482W error message The BUFFPAGE parameter will only be used if one of the buffer pools is defined with a size of -1, change the buffer pool size:

```
ALTER BUFFERPOOL buffpoolname SIZE -1
```

9. Set the SQL statement heap size to automatic:

```
update db cfg for <dbname> using STMTHEAP 12288 AUTOMATIC
```

10. From the command line, restart DB2:

```
db2stop force
db2start
```

11. Update the database configuration for the database to 200 using the following command:

```
MAXAPPLS 200 AUTOMATIC MAXLOCKS 80 AUTOMATIC
```

12. If DB2 is installed remotely, use the DB2 Client Configuration Assistant after database creation to add the database alias to the list of available databases.

Tuning DB2 database

This section describes how to do performance tuning of a DB2 database using the Performance Wizard.

You can skip this step now and do performance tuning later. If you are doing performance tuning at a later time, verify that the parameters are set to the correct values.

If your DB2 instance runs on AIX, run the Performance Wizard from a Windows system to tune performance of the instance on AIX.

Note the following while tuning DB2 performance:

- If DB2 is installed on the Content Server host and you are installing DB2 Enterprise Edition, set the target memory to 40%.
- If DB2 is installed on a different machine from Content Server and you are installing Enterprise Edition, set the target memory to 80%.
- Ensure that the `buffpage` value is at least 6000.
- If you are installing DB2 Enterprise Edition, select **More than 10 SQL statements and 60 transactions per minute**.
- If you are installing DB2 Enterprise Edition, type in four average local connections and 20 average remote applications. These numbers can be larger depending on the number of clients connecting to your repository. A production repository can have many more client applications connecting.

Running multiple Content Servers on DB2 host

If you run multiple Content Servers on the DB2 host, you might see a DB2 SQL1224N error. This can occur with multiple repositories on the host or with multiple servers that run against a single repository. To work around this, change the following parameters:

- On AIX, set `EXTSHM` to `ON` in the environment of the DB2 instance owner. You can do this in the `.cshrc` file or the corresponding system file for the different shells.

```
setenv EXTSHM ON
```

- In the DB2 environment, type this command:

```
db2set DB2ENVLIST=EXTSHM
```

DB2 repository sizes

DB2 repositories have the following size and configuration considerations:

- In a small repository on DB2, a single tablespace contains the data and indexes, and you cannot change an index tablespace.
- A small DB2 repository has an initial datafile size of 200 MB.
- In a medium or large repository on DB2, one tablespace contains the data and another tablespace contains the indexes, and you can change an index tablespace.
- A medium DB2 repository has an initial datafile size of 400 MB and an initial index file size of 200 MB.
- A large DB2 repository has an initial datafile size of 800 MB and an initial index file size of 300 MB.

Configuring internationalization settings

Content Server runs in the UTF-8 code page. Perform the following tasks before Content Server installation:

- Install the server host code page.
- Set the code page in the database.
- Set the server host locale.

The server host locale and the server code page need not be same. For example, if the host code page is set to ISO-8859_1, the host locale would typically be set to a European language (English, French, German, Italian, Portuguese, or Spanish). If the host locale is set to French, a client that connects to the Content Server without specifying a client locale is served French data dictionary labels.

If the host locale is one of the languages supported by EMC Documentum, the data dictionary information for that locale is loaded. Otherwise, the server defaults to loading the English data dictionary information. You can load additional sets of data dictionary information by modifying the `data_dictionary.ini` file. Installing additional data dictionary information can affect server performance, and EMC Documentum only supports the languages that are shipped with Content Server.

- On Windows hosts, the host locale is set in the Regional Settings dialog box.
- On UNIX and Linux hosts, the host locale is set with the LANG environment variable.

Content Server can be installed on computers that run the following operating system code pages:

- For U.S. and Western European sites, ISO-8859_1 (Latin-1)
- For Korean sites, EUC-KR
- For Japanese sites that use UNIX and Linux, EUC-JP
- For Japanese sites that use Windows, Shift_JIS
- For Chinese sites with locale zh, ms936
- For Russian with locale ru, Windows-1251

EMC Documentum Content Server Administration and Configuration Guide contains the information on locale-based configuration settings for the data dictionary files installed with Content Server.

Installing MailApp DAR

The MailApp DAR file on Content Server is required for any clients to use the changed email management features introduced in Webtop 6.8. By default, this DAR is installed by the Content Server configuration program.

Preparing for remote key management

You can upgrade repository to add remote key management (RKM) support any time.

When you create a repository, you can choose to manage the repository encryption keys locally, or to manage them with a remote key management server. For local key management, the encryption keys are encrypted and stored in the database used for the repository metadata. In releases prior to 7.0, all repositories used local key management.

For remote key management, the encryption keys are stored in a remote key management server. Beginning with release 7.0, if you have a Trusted Content Services license then you have the option of creating repositories that store the encryption keys in RSA® Data Protection Manager.

The remote key management can be enabled for all the repositories during the installation or post-installation. If you plan to use remote key management, the crypto policy key on RKM must match the key created for Content Server.

This section contains the information on configuring the RSA® Data Protection Manager (DPM) in preparation for creating a repository that uses remote key management. The DPM documentation contains the details on installing or supporting DPM.

DPM overview

When used by Content Server for remote key management, RSA® Data Protection Manager manages the encryption keys for a repository. When a repository needs a key, it requests the key from DPM and DPM passes it to the repository. DPM uses an X.509 certificate to verify the identity of the Content Server and repository requesting the key. In the DPM administration application, a repository that is using remote key management appears in the list of DPM clients.

DPM limitations

When you are planning to use DPM for remote key management, consider the following constraints:

- All the Content Servers for a repository must use the same DPM server.
- Use a low-latency connection from Content Servers to the DPM server. High-latency connections will cause performance problems and may cause timeouts.
- A repository that uses DPM cannot be a member of a federation (or use replication).
- Dump and load of encrypted file stores is not supported for repositories that use remote key management.
- Only Active key states are supported. Setting key states in DPM to Deactivated, Destroyed, or Compromised is not supported.
- The DPM server must be running and available to the Content Server host before starting. If you attempt to start a remote key management Content server, and the DPM server is not available, Content Server will fail to start.
- None of the Content Servers for a repository can be a remote Content Server (RCS). The proximity must be less than 9000.
- You cannot use global login tickets with a repository that uses remote key management.

Identities, identity groups, and key classes

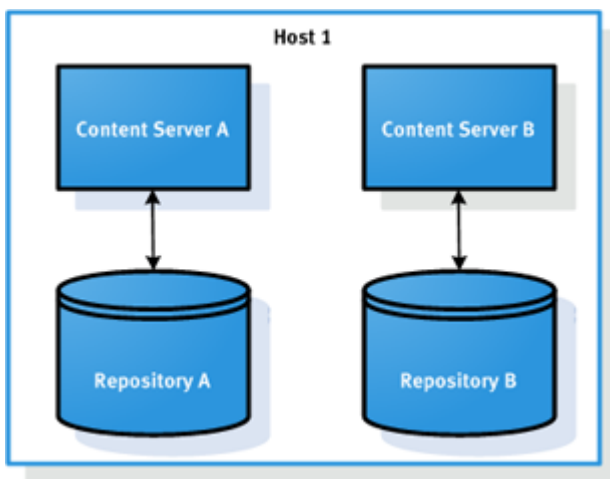
You will set up DPM to support remote key management for a repository by specifying DPM identities, an identity group, and key classes. Identities represent Content Server instances, an identity group represents a repository, and key classes represent the repository keys.

Identities

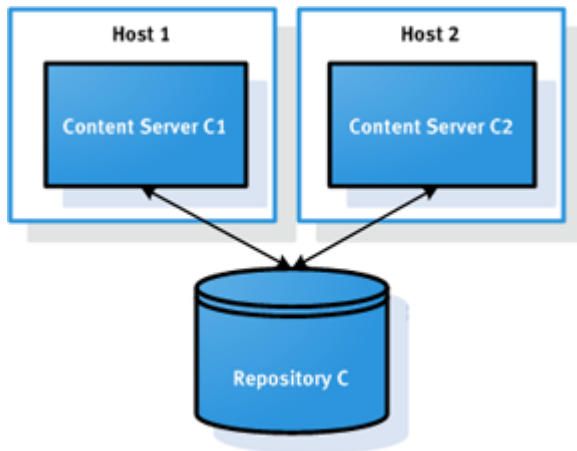
DPM uses the concept of an identity. In the context of Content Server, every Content Server instance (process) is a separate identity. In order to use remote key management, you need to configure identities in DPM and create or obtain a unique certificate for each identity. The certificate is used to authenticate a request for keys from a Content Server instance.

For example, one common scenario is to have Content Servers on a single host controlling multiple repositories. ContentServerA is one identity, and ContentServerB is a second, separate identity. If both repositories use DPM, both Content Servers must have a unique certificate.

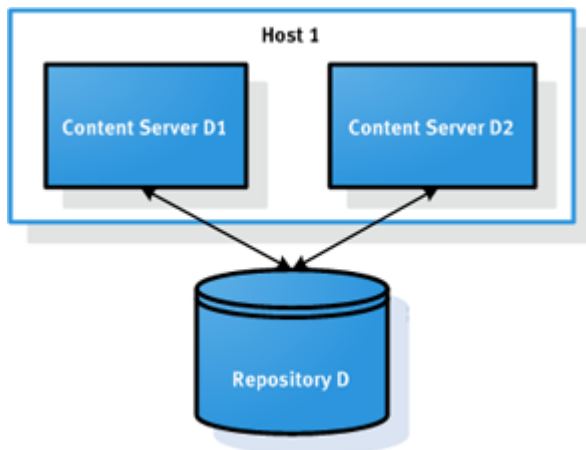
Figure 3. Single host Content Servers and multiple repositories



Another common scenario is two Content Servers on two different hosts serving a common repository. ContentServerC1 and ContentServerC2 are both serving RepositoryC. ContentServerC1 and ContentServerC2 are two separate identities. Separate, unique certificates are used to authenticate requests for keys from ContentServerC1 and from ContentServerC2, even though DPM returns the same RepositoryC keys for either request.

Figure 4. Multiple Content Servers on separate hosts for one repository

Another scenario, is to have multiple Content Servers on one host serving a common repository. ContentServerD1 and ContentServerD2 are serving RepositoryD. Again, each Content Server is a separate identity. ContentServerD1 requires a unique certificate for its identity, and ContentServerD2 requires a unique certificate for its identity.

Figure 5. Multiple Content Servers on one host for one repository

Identity groups

In the context of DPM, identity groups are a way of grouping related identities together. For remote key management using DPM, we use identity groups to keep all the identities used for a repository together. For this reason, we recommend that you name the identity group with the name of the repository it represents.

From the examples, we put the identities for ContentServerC1 and ContentServerC2 into an identity group. We'll call that identity group RepositoryC so we know it contains all the identities used for RepositoryC. We also put the identities ContentServerD1 and ContentServerD2 into the identity group RepositoryD. It contains all the identities for RepositoryD.

Key classes

In the context of remote key management for Content Server, the DPM key classes contain the keys for the repository. Each repository has its own set of keys managed by DPM. When keys are created, they are associated with an identity group and any identities in that identity group can access the keys. That is why we create an identity group for each repository, and why we associate the identities for each Content Server that serves that repository with that identity group. In that way, any Content Server that serves that repository can get the repository keys.

In the earlier examples, the key classes for RepositoryC are associated with identity group RepositoryC. The identities for ContentServerC1 and ContentServerC2 are also associated with identity group RepositoryC. Both Content Servers can access the keys for RepositoryC. In the same way, the key classes for RepositoryD are associated with identity group RepositoryD, and the two Content Server identities, for ContentServerD1 and ContentServerD2, are associated with identity group RepositoryD. Both ContentServerD1 and ContentServerD2 can access the keys for RepositoryD.

You will create four key classes for each repository to hold these four keys:

- DBK-repository key
- LTK-login ticket key
- FSK-file store key
- PPK-private/public key

There is one other key used by Content Server, the Administration Encryption key (AEK). It is the master key for Content Server. DPM does not manage the AEK. It is used to access all the other keys.

The names of the key classes in DPM must be unique, so we recommend that you name them for the repository that uses them. Never use the same key class name for different repositories.

Acquiring certificates

Each identity requires its own certificate. Before you configure a repository for remote key management, you must obtain a certificate for all the identities you need for your particular installation. You will supply the certificate when you create an identity in DPM, and supply that same certificate when you configure a Content Server that corresponds to that identity.

You will also need the root certificate for the certificate authority of the identity certificate when you configure Content Server.

Obtain the certificate from your preferred certificate authority. Generally, you will get versions of your certificate in a few different formats from the vendor. Each format uses a different file extension. When you create the DPM identity, you will use the certificate with the .cer file extension. When you configure Content Server, you will use the certificate with the .p12 extension.

Preparing DPM for remote key management

Before you begin preparing RSA® Data Protection Manager to support remote key management, decide how many identities (Content Servers) you need to create for your repository and get digital certificates for each identity.

We strongly recommend using our naming convention to manage the DPM entries:

- Identity Group—name of the repository
- Identity—concatenate the repository name and Content Server name
- Key Class—concatenate the repository name with the key name

Creating the identity group, identities, and key classes

In the following steps, substitute the repository name for *repository* and Content Server name for *content_server*.

1. Create the identity group for the repository.
 - a. From the DPM Administration application, select **Identity Groups > Create**.
 - b. Enter *repository* for **Name**.
 - c. Select **Save**.
2. Create the identities for each Content Server for the repository.
 - a. Select **Identities > Create**.
 - b. Enter *repository_content_server* for **Name**.
 - c. Select the one identity group named *repository* created earlier for **Identity Groups**.
 - d. For **Roles**, select **Operational User**.
 - e. For **Identity Certificate**, select **Browse** to navigate to the .cer file for this certificate, select it, then select **Open**.
 - f. Select **Save**.

Repeat the steps (a to f) to create any additional identities needed for this repository.
3. Create the DBK key class for the repository.
 - a. Select **Key Classes > Create**.
 - b. Enter *repository_DBK* for **Name**.
 - c. Select the identity group named *repository* created earlier for **Identity Group**.
 - d. For **Key Duration**, select **Activated Keys Have Duration**, but deselect **Get Duration From A Crypto Policy**.
 - e. Select **Next**.
 - f. For **Key Details > Cipher** select **Algorithm:** AES, **Key Size:** 128, and **Mode:** CBC.
 - g. For **Key Details > Duration** select **Duration:** Infinite.
 - h. For **Key Details > Get Key Behavior** select **Type** New Key Each Time.
 - i. Select **Next**.
 - j. For **Class Attributes**, select **Next** (do not set any class attributes).
 - k. For **Attribute Specifications**, select **Next** (do not set any attribute specifications).
 - l. Review the new key class values in the following table, then select **Finish**.

Name	Value
Name	<i>repository_DBK</i>
Identity Group	<i>repository</i>
Use Crypto Policy	False
Algorithm	AES
Key Size	128
Mode	CBC
Duration	Infinite
Get Key Behavior	New Key Each Time
Class Attributes	None
Attribute Specifications	None

4. Create the FSK and LTK key class.

Use the procedure in the previous step to create two more key classes, one named *repository_FSK*, and one named *repository_LTK*. Set all the other values for these two key classes the same as in the previous step.

5. Create the PPK key class.

- a. Select **Key Classes > Create**
- b. Enter *repository_PPK* for **Name**.
- c. Select the identity group named *repository* created earlier for **Identity Group**.
- d. For **Key Duration**, select **Activated Keys Have Duration**, but deselect **Get Duration From A Crypto Policy**.
- e. Select **Next**.
- f. For **Key Details > Cipher** select **Algorithm**: RSA and **Key Size**: 1024 (you cannot select a Mode).
- g. For **Key Details > Duration** select **Duration**: Infinite.
- h. For **Key Details > Get Key Behavior** select **Type** New Key Each Time.
- i. Select **Next**.
- j. For **Class Attributes**, select **Next** (do not set any class attributes).
- k. For **Attribute Specifications**, select **Next** (do not set any attribute specifications).
- l. Review the new key class values in the following table, then select **Finish**.

Name	Value
Name	<i>repository_PPK</i>
Identity Group	<i>repository</i>
Use Crypto Policy	False
Algorithm	RSA
Key Size	1024

Name	Value
Mode	-
Duration	Infinite
Get Key Behavior	New Key Each Time
Class Attributes	None
Attribute Specifications	None

Information required to configure repository

After you have created the identity group, the identities, and the key classes, DPM is ready to support remote key management for a repository. During repository configuration, if you have installed a TCS license, you will have the option of providing the information required for the repository to use remote key management. You will need to have the following information during repository configuration:

- DPM hostname or IP address
- DPM port number
- identity certificate for Content Server (.p12 format of the certificate)
- password for that identity certificate
- a root certificate for the Certificate Authority (CA) of your identity certificate
- the names of the four key classes you created for the repository

Preparing installation package

Download the Content Server software for your operating system and database.

- On Windows, expand the compressed archive by double-clicking the file.
- On UNIX and Linux, expand the compressed archive by typing:

```
% tar -xvf filename
```

Ensure that you have execute permission on the `serverSetup.bin` file. If not, add execute permission to the file by running this command:

```
chmod +x serverSetup.bin
```

Configuring for Certificate-based SSL communication

This section describes the information on Certificate-based SSL communication.

Connection modes

Documentum Content Server and connection broker support the following connectivity modes:

Mode	Description
Native	Connections are in the raw RPC format without any encryption
Secure	SSL mode is: <ul style="list-style-type: none"> used with anonymous ciphers (ADH:DEFAULT) used with certificates
Dual (Native & Secure)	Connections are both in the raw RPC format without any encryption and SSL mode is used with anonymous ciphers (ADH:DEFAULT) and certificates.

We strongly recommend you to configure Content Server, connection broker, and Documentum Foundation Classes (DFC) to be in the Secure mode with Certificate-based SSL.

In addition to the existing connectivity options, a new connectivity mode is available. In this mode, certifications are used to verify SSL servers.

Note: Certificate-based SSL and non-anonymous mode SSL are synonymous.

Documentum Foundation Classes (DFC) supports the following connectivity modes:

Mode	Description
Native	Connections are in the raw RPC format without any encryption
Secure	SSL mode is used with anonymous ciphers (ADH:DEFAULT)
Try_native_first	Native mode is attempted first, falling back to secure
Try_secure_first	Secure mode is attempted first, falling back to native

In the OOTB setup, the secure mode will continue to use anonymous mode by default. In the anonymous SSL mode, certificates are not used. In addition, both end parties are trusted and not verified. You must configure Content Server, connection broker, and Documentum Foundation Classes (DFC) to use Certificate-based SSL.

Note the following details about Certificated-based SSL:

- Clients verify the servers. For example, DFC verifies the Content Server during connection handshake.
- Clients will connect only to servers that are in the trust store of the SSL client.
Note: A trust store is a file containing certificates from trusted servers.
- Only SSL server verification is supported. No verification is provided to the SSL clients.

Prerequisites

EMC does not provide any certificates. You must procure or generate your certificates and manually configure the components to use the non-anonymous/Certificate-based SSL mode.

Certificates

You must purchase certificates from a Certificate Authority (CA).

Perform the following steps to procure a certificate:

1. Provide details of the certificate to the CA in the Certificate signing request (CSR) format. The CA obtains CSR from user and signs it and provides the signed certificate to the user.
2. You will receive certificate signed by a CA and a private key with CA certificate chain.
3. Depending on the CA, the certificate and chain can be in the PKCS#12 or plain DER or PEM format. PKCS#12 is widely-used specialized keystore format standardized by RSA.
 - PKCS#12 format is provided with a password
 - The private key is provided separately for PEM/DER certificates

You can also generate certificates using tools. Such certificates are known as self-signed certificates and are generally used for testing.

There is no restriction or limitation on the certificate type, private key type/length, or certificate chain length.

Building a Keystore

The keystore must be in the PKCS#12 format.

If certificates are in the PKCS#12 format, no additional configuration is required. Proceed to configure the connection broker, server, and DFC.

If the certificate and private key are provided in PEM or DER format, you must use tools to import to the PKCS#12 format and then perform the configuration.

You can use OpenSSL or Keytool, widely used SSL library or tool to generate certificates and keys, convert among the different representations, convert among the key/certificate/store formats, dump key/certificate/store details/contents, debug connections, and so on. The *OpenSSL and Keytool documentation* contains the information.

Preparing Keystore password

The PKCS#12 keystore is provided with a password for security. If the CA has provided the certificate in this format, the password will also be available.

Password formats:

- The password can be provided in a plain text format.
- The password can also be provided in an encrypted format using `dm_encrypt_password` utility that is bundled along with Content Server installation. The `dm_encrypt_password` utility uses `aekey` in `$DOCUMENTUM/dba/secure` to encrypt the password.

Note: This is similar to how the Database password is stored on the Content Server machine.

If the password is encrypted, it will be specific to the Content Server where it was encrypted. It cannot be copied to any other Content Serve that has a different `aekey`.

Building a Trust Store

Considerations for building a trust store:

- For Content Server, the trust store must be in the PKCS#7 binary format. PKCS#7 is a widely-used specialized trust store format standardized by RSA.
- For DFC, the trust store must be in the JKS format. JKS is a widely-used specialized keystore format standardized by Java.

Note: The keystore is a file containing the certificate and private key.

The trust store contains all certificates that the SSL client can trust. The certificate list includes the list of all root and intermediate CAs. It is optional to store the leaf certificates (connection broker certificate) if you obtained it from CA. The certificate chain is adequate to validate the leaf certificate.

If the certificates are self-signed and do not include a certificate chain, then all the certificates must be stored in the trust store. However, if the certificates are self-signed and include a certificate chain, then all certificates, except the leaf certificate, must be stored in the trust store.

Location of the keystore file, keystore password file, and trust store file

The connection broker keystore, connection broker keystore password file, Content Server keystore, Content Server keystore password file, and the Content Server trust store must reside in the \$DOCUMENTUM/dba/secure directory.

The DFC trust store can be stored anywhere on the local machine which is accessible to DFC.

Configuration

Configuration is a one-time activity. Subsequent Documentum patches/upgrades will not affect the configuration.

Configuring connection broker

The connection broker works as a SSL server. It requires details of the keystore and the keystore password.

Connection broker settings are stored in the `docbroker.ini` file as follows:

Parameter Name	Description
<code>crypto_keyname = <CSaek></code>	It is the AEK key name that is used to encrypt the password in the broker.pwd file.
<code>crypto_lockbox = <crypto_lockbox></code>	It is used if lockbox is present.
<code>keystore_file = <filename></code>	Keystore containing the connection broker certificate and private key

Parameter Name	Description
keystore_pwd_file = <filename>	File containing the plain text or encrypted keystore password
Cipherlist = <list>	<p>List of ciphers separated by ":". RSA library supports these ciphers.</p> <ul style="list-style-type: none"> When DFC (client) and Content Server uses bundled Java: AES128-SHA:EDH-RSA-AES256-GCM-SHA384:EDH-RSA-AES256-SHA:AES256-GCM-SHA384:AES256-SHA:EDH-RSA-AES128-GCM-SHA256:EDH-RSA-AES128-SHA:AES128-GCM-SHA256:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA When DFC (client) and Content Server uses different Java: AES128-SHA:EDH-RSA-AES256-SHA:AES256-SHA:EDH-RSA-AES128-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA

Note: The ciphers that work depends on the Java version and its capabilities.

Sample:

```
keystore_file=broker.p12
keystore_pwd_file=broker.pwd
cipherlist = aes128_sha
crypto_keyname = CSaek
crypto_lockbox = lockbox.lb
```

Note: For Content Server HA, you should use the same certificates for Content Server and connection broker from Content Server 1. Copy the certificates from primary Content Server to secondary Content Server.

Configuring Server

Content Server works both as an SSL server and an SSL client. It requires details about the keystore, keystore password, and trust store.

Content Server settings are stored in the `server.ini` file as follows:

Parameter Name	Description
keystore_file = <filename>	Keystore containing the connection broker certificate and private key
keystore_pwd_file = <filename>	File containing the plain text or encrypted keystore password
truststore_file = <filename>	File containing the trusted connection broker certificates
Cipherlist = <list>	List of ciphers

Sample:

```
keystore_file=server.p12
keystore_pwd_file=server.pwd
truststore_file=server-trust.p7b
```

```
cipherlist = aes128_sha
```

Note: Ensure that the PKCS#12 certificate file is created using FIPS-compliant `-descert:` option.

Configuring DFC

DFC works as an SSL client. It requires details about the trust store.

DFC settings are stored in the `dfc.properties` file as follows:

Parameter Name	Description
<code>dfc.security.ssl.use_existing_truststore = <boolean:on off></code>	To specify whether DFC uses the default Java trust store or a specified trust store <ul style="list-style-type: none"> On/True: Use Java's trust store Off/False: Use the provided trust store
<code>dfc.security.ssl.truststore = <filename path></code>	File containing trusted connection broker and Content Server certificates
<code>dfc.security.ssl.truststore_password = <password></code>	To specify the plain text or encrypted trust store password

Sample:

```
dfc.security.ssl.use_existing_truststore = FALSE
dfc.security.ssl.truststore = c\:/Documentum/dba/secure/dfc.keystore
dfc.security.ssl.truststore_password = password
```

Installer support

During the configuration of connection broker and Content Server, the installer will prompt you for Certificate details which will be used for configuration.

Compatibility

All the components in a setup must use a consistent secure mode. For example, the Content Server, connection broker, and DFC must be in the anonymous (default) mode or all components must be in the certificate mode.

A mixed mode where some components are in anonymous and some in Certificate-based is not supported.

When a new component, for example, another connection broker or Content Server, is installed, the new component will not have certificate settings. Therefore, you must bring all the other related components to the anonymous mode before performing the installation.

Since legacy (prior to 7.1) clients and components will not be able to communicate in the non-anonymous mode, you must use only the anonymous mode in the system that has legacy clients.

Constraints and limitations

- RSA libraries are FIPS-compliant.
- All files must be available in the \$DOCUMENTUM/dba/secure directory. This is designed so that access to this directory can be monitored or restricted for enhanced security.

Installing and configuring Content Server

Installation workflow

The Content Server installation process can be broken down into two main processes: You first run Content Server installer to copy Content Server program files from the installation media into appropriate directories on the host. You then create connection brokers and repositories in the Content Server configuration program.

When the installer completes copying Content Server program files at the end of the first process, you can either proceed by having the installer launch the configuration automatically, or exit the installer and manually launch the configuration later to resume installation. If you choose the first option, the configuration program automatically creates a connection broker using the default ports (1489 and 1490) while in the second option, you will need to set up a connection broker first using the configuration program before creating a repository.

During installation, you have the option to enter license keys for optional modules you plan to activate. If you do not activate them in the installer, you can always do so at a later time by using the configuration program.

When a module is activated, the configuration program displays some additional module-specific screens for enabling the extended feature when you configure a repository.



Caution: If you plan to use remote key management for your repository, you must activate the Trusted Content Services (TCS) module before you configure Content Server. If the TCS license is not present, you cannot create a repository that uses remote key management, and you cannot enable it at a later time.

Installing and configuring Content Server

Using GUI

You can install Content Server by running the `serverSetup.exe` installation program on Microsoft Windows or the `serverSetup.bin` installation program on UNIX and Linux.

Installing Content Server program files

1. Log on to the computer where you want to install Content Server as the installation owner.
2. Run `serverSetup.exe` (Windows) or `serverSetup.bin` (UNIX and Linux) to launch Content Server Installer.
3. Accept the license agreement and click **Next**.
4. (Only for Windows) Specify an installation directory for Content Server.
 The installation directory name must not contain spaces. For example, do not use `c:\Documentum Products` as the name of the installation directory.
 On Windows, the default installation directory is `C:\Documentum`.
 On UNIX and Linux, there is no default installation directory. You can install Content Server in `/usr/local/bin/Documentum`.
 Click **Next**.
5. On Windows, type the installation owner password.
 On UNIX and Linux, specify if you want to run the `dm_root_task` script.
 - **Run `dm_root_task` as root:** Run the `dm_root_task` script now.
 If you choose this option, you will be prompted to provide the root user password.
 - **Run `dm_root_task` manually:** Run the `dm_root_task` script manually after the installation.
 If you choose this option, you must run the `dm_root_task` script manually to set the correct file permissions on a pair of programs required for user authentication (`dm_check_password` and `dm_change_password`).
 Run the script as the root user with the root user password.
 - a. Log in as the root user.
 - b. Navigate to the `$DOCUMENTUM/dba` directory and run script: `./dm_root_task`
 - c. Type the name of the administrators group and press Enter.
 The permissions on the files are changed.
 Click **Next**.
6. Set the administrator password and specify an available listening port for the embedded application server used by Content Server:
 - **Admin User Password:** The password must *not* have the `"`, `'`, `<`, `>`, `%`, `|`, `^`, `&`, `(`, and `)` characters. The application server administrator username is set to *admin* by default.
 - **Listen Port:** The port on which the application server listens for standard administration connections. A total of 20 ports, starting from the one you specify, will be used by the application server, and all of them must be available. The default port number is 9080. Whether you accept the default port or choose another one, do not change this port after the initial configuration.
 Click **Next**.
7. Review the installation summary and click **Install** to begin installation.

8. Specify if you want to enter license keys for optional modules of Content Server and click **Next**.
 - **Yes:** In the next screen, 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.
9. Installer prompts you if you want to configure Content Server now.
 - **Configure now:** Configure now and have the installer launch the configuration program.

If you choose this option, you will *not* be prompted to choose the connection modes (**Native**, **Secure**, and **Native and Secure**). The configuration program automatically creates a connection broker in **Native and Secure** mode using the default ports (1489 and 1490) and you proceed to create a repository.

Note: Ensure that the default ports are available.
 - **Configure later:** Configure later and exit the installer. You then manually launch the configuration program to resume the installation process.

If you choose this option, you will be prompted to choose the connection modes (**Native**, **Secure**, and **Native and Secure**). You must create a connection broker first in the configuration program before creating a repository.

Creating a connection broker

Skip these steps if you chose **Configure now** in the installer, in which case the installer automatically launches the configuration program and creates a connection broker for you using the default ports (1489 and 1490).

Follow these steps if you chose **Configure later** in the installer:

1. From `$DM_HOME\install`, run `Server_Configuration_Program.exe` (Windows) or `dm_launch_server_config_program.sh` (UNIX and Linux) to launch the Content Server configuration program.
2. In the configuration options screen, choose **Connection broker** and click **Next**.
3. On Windows, type the installation owner password and click **Next**.

On UNIX and Linux, you will not be prompted for installation owner password.
4. Choose **Add a connection broker** and click **Next**
5. Choose a connection mode in which the client will connect to the connection broker:
 - **Native:** Content Server listens for client connection requests only on standard native ports. Content Server refuses requests for a secure connection.
 - **Secure:** Content Server listens for client connection requests only on a secure TLS/SSL (Transport Layer Security/Secure Sockets Layer) port for encryption. The client and Content Server do not use TLS/SSL authentication to authenticate each other. However, the information exchanged between the client and Content Server is encrypted. Content Server refuses connection requests other than TLS/SSL connections.
 - **Native and Secure:** Content Server accepts both native and secure connection requests.

Click **Next**.

6. Select **Use certificates** if you want to enable SSL certificates.

If selected, provide the following information:

- **Keystore file name:** The name of the keystore file in the PKCS#12 format.
- **Keystore password file name:** The name of the keystore password file.
- **Cipher list**

Provide the DFC trust store information:

- **TrustStore:** The location of the DFC trust store. For example, *c\:/Documentum/dba/secure/dfc.keystore*
- **Password:** The password of the trust store file.

Select **Use Default Java TrustStore** if you want to use the default DFC Java trust store.

Click **Next**.

Note: The **Use certificates** option will be visible only if you choose the connection mode as **Secure** or **Native and Secure**.

7. Select the AEK or lockbox file for **Select AEK file** from the list box.
 - a. If you select the AEK file, provide the information for **AEK key name** and **AEK passphrase**.
 - b. If you select a lockbox file, provide the information for **AEK key name**, **AEK passphrase**, and **Lockbox file name**.

Click **Next**.

8. Provide the connection broker information:
 - **Connection broker name:** A unique name for the connection broker.
 - **Connection broker port:** The port you specify and its subsequent port will be used by the connection broker and both must be available. The default port is 1489. If you accept the default port, make sure that the next port 1490 is also available.
 - **Service startup type:** On Windows, specify whether the connection broker service starts automatically at startup or is started manually. If you use the key with a passphrase, manual option only is available.
9. Review the summary information and click **Next**.
10. When the connection broker setup is complete, choose to perform additional configurations or finish configuration.

Creating a repository



Caution: Repositories using different AEK keys can only function independently. They cannot participate in a federation, replication, and so on where encrypted data or password are shared among them. If you are using multiple repositories with different AEK keys in lockbox, each repository instance should use a different JMS instance running on different port and with different `dfc.properties`, such that it points to the appropriate repository.

1. If you are not in the Content Server configuration program, from `$DM_HOME\install`, run `Server_Configuration_Program.exe` (Windows) or `dm_launch_server_config_program.sh` (UNIX and Linux) to launch the configuration program.
2. In the configuration options screen, choose **Repository** and click **Next**.
3. Type the installation owner password and click **Next**.
4. Choose **Add a new repository** and click **Next**.
5. Choose **Use existing AEK key** or **Create new or update existing AEK key** and click **Next**.



Caution: Changing the AEK key should be a planned activity since this is used for multiple purposes such as encrypting all other keys, CNT files, passwords, other servers in federation, and so on. Once the primary Content Server uses an upgraded key, all other participating servers in the system which were using the old key should also start using the new key. If the installer is used to upgrade the other servers, configuration updates and copying of key is handled by the installer. However if this is done manually, you have to modify `server.ini` and copy the key to all other systems. It is recommended to take a backup of the old AEK key and CNT files.

- a. If you choose **Use existing AEK key**, select the AEK or lockbox file for **Select AEK file** from the list box.
 - i. If you select the AEK file, provide the information for **AEK key name** and **AEK passphrase**.
 - ii. If you select a lockbox file, provide the information for **AEK key name**, **AEK passphrase**, and **Lockbox file name**.
- b. If you choose **Create new or update existing AEK key**, specify the following information:
 - i. **Select AEK algorithm**
 - ii. **AEK key name**
By default, the name is `CSaek`.
 - iii. **AEK passphrase**
If you do not specify AEK passphrase, it takes a default passphrase for the AEK to create. To upgrade a repository prior to 7.2 that is using a custom passphrase, run the `dm_crypto_boot` utility. Use the following command:

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


After running the command, invoke the installer and upgrade the repository.
 - iv. By default, **Enable Lockbox** is selected. Specify the following information:
 - **Lockbox file name**
Lockbox file name must have `.lb` as an extension.
 - **Lockbox passphrase**

Click **Next**.

6. Specify a **Data directory path** for storing content files and indicate whether it resides on a **SAN** or **NAS device**.

The data directory must not be a top-level directory on a SAN or NAS device such as `\\x.x.x.x` where `x.x.x.x` is the IP address. For SAN or NAS, enter the complete path including a shared device. Here is an example of a valid data directory on a SAN or NAS device:
`\\x.x.x.x\folder1\folder2`.

The default data directory is `$DOCUMENTUM/data`.

Click **Next**.

7. Specify a share directory for storing client applications, code examples, and libraries.

The share directory can be on the Content Server host or on another host that Content Server can access over the network.

If you specified a data directory that resides on a SAN or NAS device in the previous step, the share directory is automatically created as a sibling on the same level as the data directory. For example, if the data directory is `\\x.x.x.x\Documentum\data` where `x.x.x.x` is the IP address, the following share directory is automatically created: `\\x.x.x.x\Documentum\share`.

The default share directory is `$DOCUMENTUM/share`.

Click **Next**.

8. Type the fully qualified domain name (FQDN) of the Content Server host computer and click **Next**.

A fully qualified domain name specifies its exact location in the tree hierarchy of the Domain Name System (DNS). For example, given a device with a local hostname `myhost` and a parent domain name `example.com`, the fully qualified domain name is `myhost.example.com`.

9. Provide the following information for the repository:

- **Repository name:** The name for a repository can have up to 32 characters, and must consist of letter, numbers, or underscores (`_`). The first character must be a letter. All letters and numbers in the name must be ASCII characters. Do not include spaces or non-alphanumeric characters. The repository name `docu` is reserved by the system.
- **Repository ID:** Valid repository IDs are shipped with the Content Server software. You can also specify your own repository IDs. The repository ID can be any number from 1 to 16777215 and must not start with a zero (0). Each repository ID must be unique on the network. You can request for additional repository IDs from the EMC Online Support website to ensure that each of your repository IDs is unique.
- **Description:** Optionally, type a brief description of the repository.
- **Authentication domain:** The default domain if the user does not specify a Windows domain when connecting to the repository. Choose the domain with the largest number of users. The configuration program automatically fills in this information.
- **Service startup type:** On Windows, specify whether the repository service starts automatically at startup or is started manually. If you use the key with a passphrase, manual option only is available.

Click **Next**.

10. Provide the connection broker information and specify if you want to enable SSL certificates:

- **Connection Broker Port:** Type the connection broker port number.
- **Connection Broker Host:** Type the host name.
- **Use certificates:** Select if you want to enable SSL certificates.

If **Use certificates** is selected, provide the DFC trust store information:

- **TrustStore:** The location of the DFC trust store. For example, `c\:/Documentum/dba/secure/dfc.keystore`
- **Password:** The password of the trust store file.

Select **Use Default Java TrustStore** if you want to use the default DFC Java trust store.

Click **Next**.

11. Choose a connection mode for the repository:

- **Native:** The client connects to the repository through a non-TLS/SSL port.
- **Secure:** The client connects to the repository through a secure TLS/SSL port. The client and the repository do not use TLS/SSL authentication to authenticate each other. However, the information exchanged between the client and the repository is encrypted.
- **Native and Secure:** The repository accepts both native and secure connection requests.

If you choose **Secure** or **Native and Secure** as the connection mode, the **Use certificates** option will be visible.

Select **Use certificates** if you want to enable SSL certificates.

Note: You must select **Use certificates** if you had enabled SSL certificates while creating the connection broker.

If selected, provide the following information:

- **Keystore file name:** The name of the keystore file in the PKCS#12 format.
- **Keystore password file name:** The password of the keystore file.
- **Cipher list**

Note: The repository must use non-anonymous SSL if the connection broker uses non-anonymous SSL.

Provide the DFC trust store information:

- **TrustStore:** The location of the DFC trust store. For example, `c\:/Documentum/dba/secure/dfc.keystore`
- **Password:** The password of the trust store file.

Select **Use Default Java TrustStore** if you want to use the default DFC Java trust store.

Click **Next**.

12. Configure your relational database management system (RDBMS).

- a. Choose whether to create a new database and a repository owner (database user with the database owner role) or use existing ones.

If you choose to use the existing database, you will not be prompted for the administrator access details. The database administrator can create a database before starting the Content Server configuration program and choose to use the existing database to create a repository. However, the database administrator access details are mandatory if you choose to create a new database.

b. Type or review the following information:

- **Data source name** (for Windows) or **Net Service Name** (for UNIX and Linux): The data source used to connect to the database server.
- **Administrator name** and **Administrator password**: The database administrator account has privileges to create and delete databases and perform other database administrative tasks.
- **Database name**: The name of the database Content Server uses to store content metadata as well as system and repository information.
- **Repository Owner Account Name** and **Password**: The database user account with the database owner role that has read and write access rights to the database.

Click **Next**.

c. Configure the data file or data devices information: The configuration program automatically fills in the information for **Data device file path** and **Log device file path**.

Click **Next**.

13. (Only for Windows) Provide the SMTP information:

- **SMTP server name**: The name of the SMTP server or the IP address.
- **Owner's email address**: The installation owner's email address that Content Server will use for email notifications.

This is a required field. If a valid SMTP server host name is not available, enter an invalid one. If the configuration program fails to connect to the SMTP server you provide, a warning is displayed, but you can still proceed with the installation.

You can modify the SMTP server host name after installation.

Click **Next**.

14. Specify if you want to use the repository you just created as a global registry or to use another one:

- **Yes**: Designates the repository just created as the global registry.

Type the **Login name** and **Password** information for the global registry user in the current repository. Do not use the installation owner or the repository owner credentials.

Client applications and other repositories will use this login name and password to connect to the global registry. Record the credentials so that you can provide this information when installing other Documentum products that require global registry access.

- **No**: The **Specify whether to use another repository** screen is displayed.
 - **Yes**: Designates an existing repository as the global registry.

Type the information about the repository you want to use as the global registry:
Connection Broker Host and **Port**

Select **Use certificates** if you want to enable SSL certificates.

The current repository will be configured to access the remote global registry.

Note: You can change the global registry designation and edit connection information after installation through Documentum Administrator or the `dfc.properties` file. EMC

Documentum Content Server Administration and Configuration Guide contains the instructions on enabling a repository as a global registry.

- **No:** The global registry is not configured. You can manually add the global registry through Documentum Administrator or the `dfc.properties` file.

15. Choose the modules you want to activate for the repository.

If you chose XML Store module, specify the following:

- XML Store deployment mode to enable XML Store.
- Set the xDB superuser password.
- Provide the XML Store port and directory location.

Click **Next**.

16. Review the summary information and click **Next**.

17. Choose to perform additional configurations or finish configuration.

Always click **Finish** to exit the Content Server configuration program; otherwise, you will have to manually start the application server services.

Note: The default settings of the IP version parameters have changed in JDK 7 and attempts to set up IP socket using IPv6, even when the machine has no IPv6 networking configured. Application need to run with Java option `-Djava.net.preferIPv4Stack=true` to use IPv4 if machine is not configured with IPv6. For JMS, add `-Djava.net.preferIPv4Stack=true` in JMS startup script if IPv6 is not configured on the machine.

Viewing the configuration summary

In the configuration options screen, select **Configuration Summary** and click **Next** to view the details of key components configured on the host, such as repository, connection broker, and JMS.

The following options are not available:

- Viewing the details of the JMSN instance.
- Viewing the user accounts in repository.
- Viewing the user accounts details in each repository by default.

Viewing the version details of installed products or components

Content Server installation program records the version information of all the products or components installed in a registry file on the target host. The information includes the product or component version, patch information, build number, installation date and so on. The registry file is located at the software installation root directory `$DOCUMENTUM/DctmRegistry.xml`. There could be multiple registry files if you install multiple Documentum platform products on one target host located in a different installation root directory.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<registry>
<product id="CS" name="Content Server">
```

```

<release base-version="7.3.0000.0155" install-date="2/3/16 7:46:15 PM PST">
<patch install-date="2/3/15 7:46:15 PM PST" version="7.3.0080.0230"/>
<patch install-date="1/3/16 7:36:15 PM PST" version="7.3.0050.0219"/>
<components>
<component>
<name>dfc</name>
<version>7.3.0010.0155</version>
</component>
<component>
<name>jdk</name>
<version>1.7.0_72</version>
</component>
</components>
</release>
</product>
</registry>

```

Using command line

You can install, configure, and also uninstall Content Server from the command line. The following operations can be performed:

- Installing Content Server
- Adding new connection broker
- Upgrading connection broker
- Deleting connection broker
- Adding new repository
- Adding new server for repository
- Upgrading repository
- Deleting repository
- Uninstalling Content Server

Creating the silent installation files

1. Create the silent installation files when you are installing Content Server using the GUI by running the following commands from the command line:

For Windows:

- Content Server:

```
serverSetup.exe -r C:\silent\Windows\<Silent Installer Property file name>
```

- Connection Broker and Repository:

```
Server_Configuration_Program.exe -r C:\silent\Windows\<Silent  
Installer property file name>
```

- remote Content Server:

```
cfsConfigurationProgram.exe -r C:\silent\Windows\<Silent  
Installer property file name>
```

- Java Method Server:

```
jmsConfig.exe -r C:\silent\Windows\<Silent Installer property file name>
```

For UNIX and Linux:

- Content Server:

```
serverSetup.bin -r <Path of the Silent Installer property file  
name>/<Silent Installer property file name>
```

- Connection Broker and Repository:

```
dm_launch_server_config_program.sh -r <Path of the Silent Installer  
property file>/<Silent Installer property file name>
```

- remote Content Server:

```
dm_launch_cfs_server_config_program.sh -r <Path of the Silent Installer  
property file>/<Silent Installer property file name>
```

- Java Method Server:

```
jmsConfig.sh -r <Path of the Silent Installer property file>/<Silent  
Installer property file name>
```

2. After the files are generated, use a text editor to open the files to update or change the values of the variables, wherever required.
3. Save the configuration file.

Note: You can also use the silent installation template files provided by EMC for reference or customization. The silent installation template files are located at `$Documentum/product/<version>/install/silent/templates`.

You can create the silent installation file for remote Content Server using the silent installation file of connection broker and repository. To achieve this, you can use the silent installation tool located at `$Documentum/product/<version>/install/silent/silenttool`. The `Readme.txt` at this location contains the instructions.

Running the installation and configuration from the command line

1. Log in to the host system using your host login.
2. Change the value of `INSTALLER_UI` to ***silent*** in all the silent installation files.

Note:

- If you choose to configure the connection mode of the connection broker and the repository as **Native and Secure**, ensure that you change the value of **SERVER.DOCBROKER_CONNECT_MODE** and **SERVER.CONNECT_MODE** to ***dual***.
 - Ensure that you change the value of `common.use.existing.aek.lockbox` to ***common.create.new*** to move an existing AEK into lockbox.
3. Run the silent installation program using the following commands from the command line:

For Windows:

- Installing Content Server program files:

```
serverSetup.exe -f C:\silent\Windows\<Silent Installer property file name>
```

- Creating a connection broker and repository:

```
Server_Configuration_Program.exe -f C:\silent\Windows\<Silent  
Installer property file name>
```

- Installing remote Content Server program files:

```
cfsConfigurationProgram.exe -f C:\silent\Windows\<Silent  
Installer property file name>
```

- Installing Java Method Server program files:

```
jmsConfig.exe -f C:\silent\Windows\<Silent Installer property file name>
```

For UNIX and Linux:

- Installing Content Server program files:

```
serverSetup.bin -f <Path of the Silent Installer property file  
name>/<Silent Installer property file name>
```

- Creating a connection broker and repository:

```
dm_launch_server_config_program.sh -f <Path of the Silent Installer  
property file name>/<Silent Installer property file name>
```

- Installing remote Content Server program files:

```
dm_launch_cfs_server_config_program.sh -f <Path of the Silent Installer  
property file name>/<Silent Installer property file name>
```

- Installing Java Method Server program files:

```
jmsConfig.sh -f <Path of the Silent Installer property file name>  
/<Silent Installer property file name>
```

You may not see the progress message of the silent installation when it is run from the command line as per the InstallAnywhere design. All messages are saved in the installation log file. On UNIX and Linux, the silent installation command execution does not return until the installation is complete. However, on Windows, the silent installation command execution returns even when the silent installation is in progress. To avoid this behavior on Windows, you can launch the silent installation command using the `start /w silent-install-command` command.

Note: If an error occurs while running the installation from the command line, the installation stops. Add **KEEP_TEMP_FILE** and set it to **true** to ensure that the temporary files are retained for troubleshooting, if needed.

Uninstalling Content Server from the command line

1. Update the `linux_install.properties` or `win_install.properties` file with the following content only:

```
INSTALLER_UI=silent
```

2. Run the following command from the command line:

For Windows:

```
%Documentum%\uninstall\server>Uninstall.exe -f <Path of the Silent Installer  
property file name>/<Silent Installer property file name>
```

For UNIX and Linux:

```
$DOCUMENTUM/uninstall/server>./Uninstall -f <Path of the Silent Installer  
property file name>/<Silent Installer property file name>
```

Supporting Docker for Content Server

Introduction

You can install and configure Content Server on the supported Docker containers.

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of Operating System level Virtualization. Docker uses resource isolation features of the Linux kernel such as cgroups and kernel namespaces to allow independent containers to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines. Docker is a tool that can package an application and its dependencies in a virtual container that can run on any Linux server.

Supported Docker configurations

The Docker support for Content Server is available only on the following:

- Operating system: Red Hat Enterprise Linux, CentOS, and Ubuntu
- Database: Oracle and PostgreSQL

EMC provides the Docker images with the following configuration: Ubuntu/PostgreSQL and CentOS/PostgreSQL. You can create the Red Hat Enterprise Linux/Oracle Docker image as described in the [Creating the Content Server Linux/Oracle Docker image, page 137](#) section.

Installing Docker

1. Check your kernel version. Docker is supported on Red Hat Enterprise Linux 7 or higher. It requires a 64-bit installation regardless of your Linux version and your kernel must be 3.10 or higher. To check your current kernel version, open a terminal and run the `uname -r` command to display your kernel version.
2. Log in with root account and install the Docker Engine. Run the `$yum install docker-engine` command.
3. Start the Docker daemon service. Run the `$service docker start` command.

The *Docker Documentation* contains more information.

Common notes

- For internal applications, use the internal connection broker (for example, running on 1489). Do not have any translations. Point `dfc.properties` of internal clients to use internal Docker IP. For external applications, use the external connection broker (for example, running on 1689). Translations are done by the Docker scripts automatically. The translation is from internal Docker IP to external IP. Point `dfc.properties` to external IP of the connection broker.
- For asynchronous write and pre-caching operations in a Docker environment, perform the following:
 - Create the DMS configuration having `message_post_url`, `message_consume_url` with the internal IP (for example, `http://172.17.0.1:8489/`).
 - Change the following in `dms.properties`:
 - Provide the external IP for `dms.webservice.update.url` (for example, `dms.webservice.update.url = http://10.31.86.166:8489`).
 - Provide the internal IP for `dms.jmx.host` (for example, `dms.jmx.host = 172.17.0.1`).
- If your database is PostgreSQL, perform the following:
 - Linux: Log in as a postgres user and create a folder called `db_<RepositoryName>_dat.dat` in `/var/lib/pgsql/9.4/data/`.
 - Windows: Log in as a postgres user and create a folder called `db_<RepositoryName>_dat.dat` in `C:\Program Files\PostgreSQL\9.4\data\`.
- If you use remote file system with netshare plugin, then ensure to install the respective nfs or cifs RPMs in Docker host machine. For example: **yum install nfs*** (for NFS) and **yum install cifs*** (for CIFS).
- If you use netshare plugin for remote data and for any reason if the service is restarted, you must run the `statelesscs_config.sh` or `hacs_config.sh` script to start the container.
- If the image is a TAR file, then load the image into the local registry using the following command and update the Content Server image name: `#docker load -i <filename of the tar image>`

Creating the Content Server Linux/Oracle Docker image

Prerequisites

You must have working knowledge of Docker, Red Hat Enterprise Linux, and Oracle. In addition, you must have administrative privileges on the machine where you are installing Content Server and also have database administrator account for the Oracle server. You need two machines with the following configuration.

Hardware requirements (Machine 1 — Container for Content Server)

Item	Requirement
Operating system	Red Hat Enterprise Linux 6.7 or 7.0 (64-bit)
Free disk space	80 GB
RAM	8 GB
Swap space	8 GB
Free space in temporary directory	2 GB

Hardware requirements (Machine 2 — Container for Oracle server)

Item	Requirement
Operating system	Red Hat Enterprise Linux 7.0 (64-bit) or Windows Server
Database	Oracle server 12c

Note: Oracle server can be on any supported operating system. However, for illustrative purposes, all information in this document are provided considering Oracle server is installed on a Linux platform. The *Oracle documentation* contains more details on hardware and software requirements on this machine.

Software requirements

The [EMC E-LAB Interoperability Navigator](#) contains the software requirements information for your product.

Configuring the Red Hat Enterprise Linux base image

1. Download or pull the Docker image. For example:

```
$docker pull rhel7
```
2. Create a container to install minimum RPM Package Managers (RPMs) and Oracle client. For example:

```
$docker run -ti --name rhel7ora rhel7 /bin/bash
```
3. Copy all the packages or RPMs to the Docker container to install all the required RPMs. First, copy the packages from CD/ISO to your host Docker Machine. After that, copy the packages folder from the Docker Machine to the Docker container. For example:

```
$docker cp Packages rhel7ora: /
```
4. Log in to the Docker container and install the `createrepo` package. For example:

```
$ cd /Packages && rpm -ivh libxml2-python*  
deltarpm* python-deltarpm* createrepo*
```

5. Build the local repository in the Docker container. For example:

```
$ createrepo -v /Packages
```

6. Create the repository file in the Docker container. For example:

```
$ vi /etc/yum.repos.d/rhel7.repo
```

Add the following lines in the repository file:

```
[rhel7]
name=RHEL 7
baseurl=file:///Packages
gpgcheck=0
enabled=1
```

7. Install `gnome-packagekit` and `xeyes` RPMs to support GUI installation of Content Server in the container. For example:

```
$ yum install gnome-packagekit*
$ yum install xeyes*
```

8. Install `rng-tools` to support random number generation on Linux. For example:

```
$ yum install rng-tools
```

9. Install Oracle client-related RPMs. For example, run the following commands in the given sequence:

```
$ yum install ksh
$ yum install binutils*
$ yum install elfutils-libelf-0.*
$ yum install glibc-2.*
$ yum install glibc-common-2.*
$ yum install libaio-0.*
$ yum install libgcc-4.*
$ yum install libstdc++-4.*
$ yum install make-3.*
$ yum install compat-libcap1*
$ yum install gcc-4.*
$ yum install gcc-c++-4.*
$ yum install libaio-devel-0.*
$ yum install libstdc++-devel-4.*
$ yum install unixODBC-2.*
$ yum install unixODBC-devel-2.*
$ yum install libXtst
$ yum install sysstat*
$ yum install csh*
$ yum install hostname wget iputils
```

10. To ensure that `dmdbtest` does not fail during the loading of the shared libraries of `libsasl2.so.2`, while configuring the repository, perform the following steps in RHEL docker container:

```
root:~ # cd /usr/lib64
root :/usr/lib64# ln -s libsasl2.so.3 libsasl2.so.2
```

11. After installing all the required RPMs, remove the `/Packages` folder in the container.

Installing Oracle client

Install the Oracle client on the Docker container to connect the Oracle database which is outside of the container (in a different machine).

1. Create Oracle user and groups for Oracle client.

- a. Log in to the Docker container with root account and create groups. For example:

```
$groupadd oinstall
$groupadd dba
```
- b. Create an Oracle user with the initial login group of oinstall, secondary to dba. For example:

```
$useradd -g oinstall -G dba -s /bin/bash -d /home/oracle -m oracle
```
- c. Set the password for the Oracle user. For example:

```
$passwd oracle
```
2. Create directories for Oracle client in the Docker container. For example:

```
$mkdir -p /u01/app/oracle
```
3. Change the ownership and permissions for the directories. For example:

```
$chown -R oracle:oinstall /u01/app/oracle
$chmod -R 775 /u01
```
4. Copy the Oracle client installer from Docker to the Docker container. For example:

```
$docker cp ora_client rhel7ora:/u01/app
```
5. Log in to the Docker container with root account and change the permissions for the ora_client folder. For example:

```
$chmod -R 777 /u01/app/ora_client
```
6. Set the DISPLAY environment variable to install the Oracle client in GUI mode. For example:

```
$export DISPLAY=10.30.87.106:0.0
```
7. Navigate to the /u01/app/ora_client/ folder and run the Oracle client installer. For example:

```
$/runInstaller
```
8. After the installation, set the Oracle and path environment variables. For example:

```
$export ORACLE_HOME=/u01/app/oracle/product/12.1.0/client_1
$export PATH=$ORACLE_HOME/bin:$PATH
$export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Run **netca** to configure the Local Net Service Name. Also, provide the Oracle service name and database hostname details. Use 10.31.71.131 as Oracle hostname in the \$TNS_ADMIN/tnsnames.ora. The value (10.31.71.131) is automatically replaced by Documentum Docker scripts with the original Oracle hostname in the configuration file.
9. After configuring the Oracle client, remove the client installer /u01/app/ora_client and save the changes to the image. For example:

```
$docker commit -m "Base configuration of RHEL with Oracle client"
rhel7ora contentserver/rhelora/stateless/cs:base
```

Configuring to create the Content Server Linux/Oracle Docker image

1. Download the Content Server product binaries and place them in a temporary local FTP server.
2. Download the Documentum Docker scripts including the Dockerfile-RhelOra_Statelesscs Docker file.

3. Modify the `Dockerfile-RhelOra_Statelesscs` file with proper entries for base image and temporary Content Server build FTP server details. For example:

Base image name:

```
FROM contentserver/rhelora/stateless/cs:base
```

FTP server details:

```
wget --ftp-user=anonymous --ftp-password=password
ftp://10.8.2.186/Builds/Content_Server/
${PRODUCT_MAJOR_VERSION}/${BUILD_NUMBER}
/Server/linux_ora/*
```

4. Run the following command to create the Content Server Linux/Oracle Docker base image. For example:

```
$docker build --build-arg PRODUCT_MAJOR_VERSION=7.3
--build-arg BUILD_NUMBER=7.3.0000.0201_unverified
--build-arg INSTALL_OWNER_USER=dmadmin
-f Dockerfile-RhelOraCS_Statelesscs
-t contentserver/rhelora/stateless/cs:7.3.0000.
```

Exporting environment variables for storing passwords to Docker environment

EMC provides passwords as environment variables. Ensure that you provide valid password set on the environment variables before creating or restarting the containers. Otherwise, the installation may fail.

Export the following environment variables on the shell manually before running the Docker scripts (`statelesscs_config.sh`, `hacs_config.sh`, and `seamless_config.sh`) before restarting or recreating the container:

```
export APP_SERVER_ADMIN_PASSWORD=<web application server administrator password>
export INSTALL_OWNER_PASSWORD=<install owner password>
export ROOT_USER_PASSWORD=<root user password>
export REPOSITORY_PASSWORD=<repository password> (only for stateless configuration)
export EXTERNALDB_ADMIN_PASSWORD=<external database administrator password>
export BOF_REGISTRY_USER_PASSWORD=<bof registry user password>
export AEK_PASSPHRASE=<aek passphrase>
export LOCKBOX_PASSPHRASE=<lockbox passphrase>
```

Note: Repository passwords must not contain special characters other than the following characters: . (period), _ (underscore), - (hyphen)

Installing and configuring Content Server on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.
3. Provide all the required details in the `statelesscs.conf` file. Read the description of every field and provide valid values for each parameter.

4. Export the environment variables as described in the [Exporting environment variables for storing passwords to Docker environment, page 141](#) section.
5. Run the `statelesscs_config.sh` script.
6. To verify the installation, check the logs at `/opt/dctm_docker/logs/<hostname>.log` inside the container.

Upgrading Content Server using seamless upgrade method

Seamless upgrade from Content Server 7.1, 7.2, and 7.3 versions is supported. The *EMC Documentum System Upgrade and Migration Guide* contains the information.

Installing and configuring of Content Server HA on Docker environment

1. Install the supported version of Docker, Docker compose file, and netshare plugin in your host machine.
2. Start the Docker process from the service. For example:

```
service docker start
```
3. Start the Docker netshare plugin. For example:

```
./docker-volume-netshare --basedir=/var/lib/docker/volumes --verbose=true nfs
```
4. Share the `$DOCUMENTUM/data` and `$DOCUMENTUM/share` if the existing Content Server does not use the remote file system for data.
5. Provide all the required details in `hacs.conf` file. Read the description of every field and provide valid values for each parameter.
6. Export the environment variables as described in the [Exporting environment variables for storing passwords to Docker environment, page 141](#) section.
7. Run the `hacs_config.sh` script.
8. Add the secondary server details in `dfc.properties` of primary server. Also, add the primary server details in `dfc.properties` of secondary server. For example:

`dfc.properties` of primary server:

```
dfc.docbroker.host[3]=<IP address of the secondary server>  
dfc.docbroker.port[3]=1689
```

`dfc.properties` of secondary server:

```
dfc.docbroker.host[3]=<IP address of the primary server>  
dfc.docbroker.port[3]=1689
```

Note: Port number must be 1489 and IP addresses must be IP addresses of the containers if both the primary and secondary servers are run in same host machine.

Then, restart all services in primary server and secondary server.

9. To verify the installation, check the logs at `/opt/dctm_docker/logs/<hostname>.log` inside the container.

Completing the installation

Reviewing Content Server installation and configuration logs

Content Server Installer and configuration program both create log files. The log files are stored in one of the following directories:

- The installation owner's desktop (Windows) or user home directory (UNIX or Linux)
- The current working directory.

For Installer, the current working directory is the directory from which you started the program. For the configuration program, the current working directory is typically `$DM_HOME\install` (Windows, UNIX, and Linux).

- The parent directory of the installation directory, if the installation owner does not have write permission on the current working directory.
- The user's home directory, if the installation owner does not have write permission on the parent directory.

The log filenames are `install.log` and `UniversalServerConfigurator_Install_mm_dd_yyyy_hh_mm_ss.log`.

Each script that runs during repository configuration creates a log file. These are stored in the `$DOCUMENTUM/dba/config/repository_name` directory.

Content Server stores other log files in the `$DOCUMENTUM/dba/log` directory. After you install or upgrade Content Server, examine the log file for the repository for error reports. The log is called `repository_name.log.save.date.time`. `repository_name` is the name of the repository you created or upgraded, and `date` and `time` are the date and time the log was saved.

Configuring symbolic link path for Java

During Content Server installation, a symbolic link path for Java is installed at `$DOCUMENTUM/java64/JAVA_LINK`.

The symbolic link path points to the actual Java path and is independent of the specific Java version. The symbolic link path is used in Content Server scripts to specify the actual Java executable path. You only need to change the target of the symbolic link and need not update the scripts after the Java upgrade. The symbolic link is created on Linux and Solaris. On Windows, it is only available on the systems that support `mklink` and NTFS symbolic link.

Note: The symbolic link is not created on AIX.

Enabling the purge audit job

The purge audit job deletes old audit trail objects from the repository. The job runs as the installation owner. However, when a repository is created, the installation owner is not granted sufficient extended privileges to run the job.

After you create a repository, create a new user with superuser privileges, connect as that user, and grant the installation owner account Purge Audit extended privileges.

Post-installation tasks

After you have successfully installed Content Server, perform the tasks described in this section to configure the system as required and get it up and running.

Configuring WildFly for SSL

Install the connection broker and Content Server in *secure* or *native & secure* mode. Once the Content Server installation is complete, stop all the services (Content Server, connection broker, and JMS) and proceed with configuration for Certificate-based SSL manually.

Note: Content Server installer does not support automatic configuration.

1. Build a keystore for JMS. The keystore must be in the JKS format. A new certificate must be generated by exporting the keystore file. Ensure that the JMS certificate and the connection broker certificate are imported into the SSL client's trust store.

2. Modify the `standalone.xml` in

```
%WildFly_HOME%\server\DctmServer_MethodServer\configuration\standalone.xml
```

For example, in `standalone.xml`:

- Add the following under `<security-realms>`:

```
<security-realm name="UndertowRealm">
  <server-identities>
    <ssl>
      <keystore path="c:/keystore/jms.keystore"
        keystore-password="Password@123" alias="jmskey"
        key-password="Password@123"/>
    </ssl>
  </server-identities>
</security-realm>
```

- Add the following after `<server name="default-server">`:

```
<http-listener name="default" socket-binding="http"
  redirect-socket="https"/> :
<https-listener name="https" socket-binding="https"
  security-realm="UndertowRealm"
  enabled-cipher-suites="<cipher-list"/>
```


Use the cipher-suite values depending on the following:

- If you want to configure JMS in non-anonymous SSL mode, then cipher suite parameters are:
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - AES256-SHA256
 - DHE-DSS-AES256-SHA256
 - DHE-DSS-AES256-SHA

Note: To use AES256 and SHA256 algorithms, ensure that you have the JCE Unlimited Strength Jurisdiction policy files.

- If you want to configure JMS in dual mode (anonymous and non-anonymous), then cipher suite parameter is TLS_DH_anon_WITH_AES_128_CBC_SHA,SSL_DH_anon_WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA.

3. Dump the `dm_jms_config` object and change the `base_uri` to point to the new URL:

```
API> retrieve,c,dm_jms_config
.....080004d38000b1a
API> set,c,l,base_uri[0]
API> https://<content server host>:9082/DmMethods/servlet/DoMethod
API> save,c,l
```

4. Start the JMS.

The JMS starts successfully.

Supported communication configurations

This table contains the information on the supported communication configurations:

Content Server/Connection Broker/DFC	JMS	Client
Non-anonymous	Non-anonymous	Non-anonymous
Anonymous	Dual	Non-anonymous

In case of a combination of JMS:

- JMS1 and JMS2 can be configured for anonymous communication with Content Server.
- JMS1 and JMS2 can be configured for non-anonymous communication with Content Server.
- When using client such as Documentum Administrator, if JMS is configured for non-anonymous SSL communication with Content server, you can find the `dfc.security.ssl.use_existing_truststore` parameter in `dfc.properties`. If set to true, import all certificates (Content Server, connection broker, and JMS) in default Java trust store on the client. If set to false, provide the `dfc.security.ssl.truststore` and `dfc.security.ssl.truststore_password` parameters in `dfc.properties` on the client and import all certificates (Content Server, connection broker, and JMS) in DFC trust store.

Note:

- The Distributed Content components (clients such as ACS, BOCS, and DMS) in anonymous mode is not supported.
- The following JMS and SSL mixed mode configurations are not supported:
 - JMS1 in non-anonymous SSL mode, JMS2 in anonymous SSL mode, and Content server in anonymous SSL mode.
 - JMS1 in non-anonymous SSL mode, JMS2 in both (anonymous and non-anonymous) SSL mode, and Content server in non-anonymous SSL mode.
 - JMS1 in non-anonymous SSL mode, JMS2 in native mode, and Content server in non-anonymous SSL mode.

Changing the connection mode from native to secure

You can change the connection mode from a native to a secure connection to enable SSL encryption for traffic between DFC and Content Server. Follow the instructions in this section to change the secure connection mode from native to secure connection to enable SSL at the Content Server level.

1. Update the `dm_server_config` object:

- a. Connect to IAPI as installation owner.

- b. Run the following commands:

```
API> retrieve,c,dm_server_config
(This command retrieves the object ID.)
API> dump,c,l
(Verify that the secure_connect_mode has changed to
native by viewing the list.)
API> set,c,<objectID>,secure_connect_mode
SET> secure
API> save,c,l
API> dump,c,l
(Verify that the secure_connect_mode has changed to
secure by viewing the list.)
```

2. Update the `dfc.properties` file.

- a. Navigate to the `$DOCUMENTUM/config` directory
- b. Back up the original `dfc.properties` file.
- c. Change the entry `dfc.session.secure_connect_default=try_native_first` to:
`dfc.session.secure_connect_default=secure`.
- d. Save the `dfc.properties` file.

3. Update the `docbroker.ini` file.

- a. Navigate to the `$DOCUMENTUM/dba` directory.
- b. Back up the original `docbroker.ini` file.
- c. Change the entry `secure_connect_mode=<blank>` to:
`secure_connect_mode=secure`
- d. Save the `docbroker.ini` file.

4. Restart the Content Server.
 - Restart Documentum Master Service (from Services)
 - Wait until all Documentum services have started, (repository, connection broker, Java Method Server and Master Service).
5. Verify that the connection broker (docbroker) and repository (docbase) are connecting to a secure port.
 - a. Navigate to the \$DOCUMENTUM/dba/logs directory.
 - b. View the connection broker and repository logs and verify that they have information about secure port 1490.

Backing up keystore files

After you install the Content Server and repository, back up the keystore files, which are all the files in the directory \$DOCUMENTUM/dba/secure.

Take special care to back up and save the `aek.key` file in that directory. If the key becomes corrupted and you do not have a backup, your repository cannot be started, and you cannot access encrypted files.

When using Remote Key Management, the external keystore must also be backed up so that it can be restored in event of a system failure or data corruption. Backing up the external keystore is a best practice and must be done in addition to backing up the `aek.key` file. The RKM product documentation contains the instructions and details on creating a backup of the external keystore.

Changing the default passphrase

During Content Server installation, a keystore is created that contains a passphrase that is used for encryption. After installation, you can change the default passphrase to a custom passphrase. If you create a custom passphrase after Content Server installation, any time you restart the server host, you need to run the `dm_crypto_boot` utility. *EMC Documentum Content Server Administration and Configuration Guide* contains the instructions and details on encryption, keystores, and passphrases.

Binding Content Server to a network card

To configure Content Server to use a different network card, create an initialization file for the connection broker. The file must include a `[DOCBROKER_CONFIGURATION]` section to identify the IP address of the network card. Use the following format:

```
[DOCBROKER_CONFIGURATION]
host=IP_address_string
service=service_name
port=port_number
```

IP_address_string is the IP address of the network card.

The service name is the connection broker's service name, defined in the host's services file. The port number is the port defined in the service.

- If you include a service name, the connection broker starts by using that service.
- If you include a port number, the connection broker starts by using that port.
- If you do not include a service name or a port number, the connection broker uses the default port number 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.

The *EMC Documentum Administrator Online Help* contains the information on binding Content Server to a network card.

Installing Content Server client

Install the following Content Server client applications:

- Documentum Administrator, the primary web-based administrative client tool for configuring and administrating Content Server and repositories
- Documentum Webtop, a web-based client application for accessing and managing content in the repository

Installing Content Server using an existing database account

When you choose the option to use an existing database account while configuring repository, you can proceed only if you have a Database Administrator account.

If you cannot provide the Database Administrator account at this stage, you can perform these actions after the repository configuration is complete:

- Create the Data Management System (DMS) user in the database instance.
- Grant the DMS user permissions to access the DMS schema tables.

Enabling xPlore search for new emails

To enable the xPlore search for emails imported using `MailApp.dar`, perform the following:

1. Run the following query in the repository:

```
ALTER ASPECT mdmo_message_aspect FULLTEXT SUPPORT ADD ALL
```

2. Stop the xPlore service in the index machine.
3. Clear the BOF cache at `<xPlore Home>\<WildFly version>\server\DctmServer_Indexagent\data\Indexagent\cache\content_server_version\bof\repository_name`.

4. Start the xPlore service in the index machine.
5. Start the indexing using Documentum Administrator.

Using the migration utility

The migration utility is used for the following:

- Changing the repository ID
- Changing the repository name
- Changing the server configuration name
- Changing the hostname of the Content Server machine
- Changing the installation owner

The following files are available in the `$DM_HOME\install\external_apps\MigrationUtil` folder:

- `MigrationUtil.jar`
- `config.xml`
- `MigrationUtil.sh` and `MigrationUtil.csh` (UNIX and Linux)
- `MigrationUtil.bat` and `WinOSService.dll` (Windows)

Ensure that you perform the following before running the migration utility:

- Take a complete backup of Content Server installation manually. Also, for Windows, take a backup of the registry files. The migration utility, specifically changing the repository ID is database intensive. Ensure that you take a backup of the database manually.
- Download all the required JDBC drivers in your file system. (For example, `ojdbc.jar` (Oracle) and `sqljdbc.jar` (SQL Server))
- Append the class path location of the JDBC drivers in the `MigrationUtil.bat` (Windows) or `MigrationUtil.sh` or `MigrationUtil.csh` (UNIX and Linux).
- Stop all Documentum services such as the connection broker, repository, JMS, and so on.
- In `config.xml`, provide the inputs for the following parameters:
 - `dbms`: Database on which utility is run.
 - `tgt_database_server`: IP address of the target database server.
 - `port_number`: Listening port number of the database.
 - `InstallOwnerPassword`: Password of the installation owner.
 - `DocbaseName.1`: Name of the existing repository. If you have more than one repository, provide the list of repository names.

- `DocbasePassword.1`: Password of the existing repository owner. If you have more than one repository, provide the list of repository owner passwords.
- `isRCS`: Set to **yes** if you are running the migration utility on the secondary Content Server. Otherwise, set to **no**.

The migration utility must be run on the target Content Server host after the successful Content Server migration from source to the target version.

Note:

- The migration utility supports Content Server only.
- The migration utility does not support custom types created on OOTB Content Server installation except for using the utility for changing the repository ID. Administrator must update the custom types that has any references to repository name, server configuration name, hostname, and installation owner.
- After running the migration utility, the log files are generated in the `MigrationUtil_logs` folder. Review the log files and ensure that there are no errors. In case of errors, try and fix it manually. If you are not able to fix the error(s) manually then restore the backup version of databases and Content Server, fix the errors, and then run the utility again.

Changing the repository ID

The migration utility facilitates the change in `Repository ID` of each repository in the Content Server installation. You can change only one Repository ID for each time the utility is run. Running the utility is both time and space consuming. For example, if you have 10 million objects you may need a 50 GB disk space because of large database transactions. With the change in repository name, hostname, and installation owner along with this, the entire process results in cloning of the existing repository into another. Each repository runs as an independent identity. The change to Repository ID affects all metadata associated with the repository objects.

In the `config.xml` file, provide the inputs for the following parameters:

- `ChangeDocbaseID`: Set to **yes**.
- `Docbase_name`: Same name as specified for `DocbaseName.1`.
- `NewDocbaseID`: New Repository ID.

Changing the repository name

The migration utility facilitates the change in `Repository Name` of each repository in the Content Server installation.

In the `config.xml` file, provide the inputs for the following parameters:

- `ChangeDocbaseName`: Set to **yes**.
- `NewDocbaseName.1`: Provide an unique repository name.

Note:

- You can change the repository name for multiple repositories. For example, provide new name for `NewDocbaseName.2` for the corresponding `DocbaseName.2`.
- For Kerberos SSO, after changing the repository name, you must add the following entries in `dfc.properties`:

```
dfc.kerberos.docbaseName[0]=<new repository name>
dfc.kerberos.mapped_docbaseSPN[0]=<SPN>
```

Changing the server configuration name

The migration utility facilitates the change in `Server Configuration Name` of each repository in the Content Server installation.

In the `config.xml` file, provide the inputs for the following parameters:

- `ChangeServerName`: Set to **yes**.
- `NewServerName.1`: Provide a new server name.

Note: You can change the server configuration name for multiple repositories. For example, provide new name for `NewServerName.2` for the corresponding `DocbaseName.2`.

Changing the hostname of the Content Server machine

The migration utility facilitates the change in `hostname` of the Content Server machine with multiple repositories.

Ensure the following before running the migration utility:

- Change the hostname of the Content Server machine.
- List all the repositories of Content Server in `config.xml`.

In the `config.xml` file, provide the inputs for the following parameters:

- `ChangeHostName`: Set to **yes**.
- `HostName`: Hostname of the Content Server machine you manually changed as a prerequisite step before running the migration utility.
- `NewHostName`: Name of the new hostname you changed.

Changing the installation owner

The migration utility facilitates the change in `Installation Owner` of the Content Server installation with multiple repositories.

Ensure the following before running the migration utility:

- Create new installation owner account with similar privileges as the existing installation owner.
- Content Server is installed in a common location accessible to all with relevant permissions.
- List all the repositories of Content Server in `config.xml`.

In the `config.xml` file, provide the inputs for the following parameters:

- `ChangeInstallOwner`: Set to **yes**.
- `InstallOwner`: Name of the existing installation owner from `server.ini` of the repositories.
- `NewInstallOwner`: Name of the new installation owner you created.
- `NewInstallOwnerPassword`: Password for the new installation owner.

After changing the installation owner, run `$DOCUMENTUM/dba/dm_root_task` as root.

Note:

- Changing the installation owner for domain user does not provide the permission for the documentum folder. Run the following commands manually:

```
icaccls <Documentum>/grant <domainuser>:(OI)(CI)F /t /q
icaccls C:\Documentum /grant <domainuser>:F /t /q
```
- Perform the following steps manually to change member repository references in the host to successfully run the federation job:
 1. Change `r_member_docbases` attribute of `dm_federation` object.
 2. Rename `.cnt` files for member repositories at the `%DOCUMENTUM%\dba\config\<governing_docbase>*.cnt` location.
 3. Change object name of jobs referring to older member repositories and Master and replica-docbases in `method_arguments`. For example:

```
<select r_object_id,object_name from dm_job
where object_name like 'dm_ACLRepl%*>
```
 4. Change references of member repositories in `federation.cnt` at `*%DOCUMENTUM%\dba\config\<governing_docbase>`.

Running the migration utility

For Windows, run the `MigrationUtil.bat` file.

For UNIX and Linux, run the `MigrationUtil.sh` or `Migration Util.csh` file.

Getting Started with Content Server

Starting/stopping Content Server on Windows

On Windows systems, you start and stop connection brokers and repositories from the Server Manager tool.

To start Content Server on Windows

1. From the Windows **Start** menu, choose **All Programs > Documentum > Documentum Server Manager**.
2. Under the **DocBroker** tab, choose a connection broker and click **Start**.
3. Under the **Repository** tab, choose a repository and click **Start**.
4. Start the application server service.
 - a. Click **Start > All Programs > Administrative Tools > Services**.
 - b. In the **Services** dialog box, scroll to **Documentum Java Method Server**.
 - c. Right-click **Documentum Java Method Server** and click **Start**.

To stop Content Server on Windows

1. Stop the application server service.
 - a. Click **Start > All Programs > Administrative Tools > Services**.
 - b. In the **Services** dialog box, scroll to **Documentum Java Method Server**.
 - c. Right-click **Documentum Java Method Server** and click **Stop**.
2. From the Windows **Start** menu, choose **All Programs > Documentum > Documentum Server Manager**.
3. Under the **Repository** tab, choose a repository and click **Stop**.
4. Under the **DocBroker** tab, choose a connection broker and click **Stop**.

Starting Content Server on UNIX and Linux

To start Content Server on UNIX and Linux

1. Start the connection broker by running `$DOCUMENTUM/dba/dm_launch_docbrokerName`, where *docBrokername* is the name of the connection broker.
2. Start the repository by running `$DOCUMENTUM/dba/dm_start_serverconfigname`, where *serverconfigname* is the object name of the Content Server server config object, which consists of the host name and the repository name.
3. Start the application server by running
`$DOCUMENTUM/<WildFly version folder>/server/startMethodServer.sh.`

To Stop Content Server on UNIX and Linux

1. Stop the application server by running
`$DOCUMENTUM/<WildFly version folder>/server/stopMethodServer.sh.`
2. Stop the repository by running `$Documentum/dba/dm_shutdown_serverconfigname`, where *serverconfigname* is the object name of the Content Server server config object, which consists of the host name and the repository name.
3. Stop the connection broker by running `$Documentum/dba/dm_stop_docbrokerName`, where *docBrokername* is the name of the connection broker.

Java Method Server for High-Availability

Overview

Java Method Server (JMS) is a customized version of an application server for executing Content Server Java methods. EMC Documentum provides a servlet called DO_METHOD to execute Documentum server methods. The method server itself is a Java-based web application. It communicates with the Content Server via HTTP calls. Each time a method is invoked, the Content Server makes an HTTP request passing the name of the Java class which implements the method along with any specified arguments to a servlet which knows how to execute the specified method. JMS supports high-availability (HA). JMS supports two HA types: failover and load balancing.

- Failover

In a failover setup, if one of the JMS fails, the other JMS in the failover setup continues the method running activity from the beginning from the same Content Server.

Note: JMS that is set for failover is considered for new method execution also only when the JMS that is set for load balancing is down.

- Load balancing

If you configure JMS load balancing, method executes in a round robin manner per Content Server instance.

Installation of additional JMS

You can also install additional private JMS instances. Before you can install additional JMS instances, you must package all the web applications previously deployed to the default embedded WildFly with the `jmsPackage.bat` batch file. You can then deploy that file to the new JMS to maintain consistency between the two JMS instances.

Perform the following instructions to install and configure a second JMS on a multi-server-single-repository-single-machine configuration:

1. Ensure that only one repository is listed in the connection broker. Connection broker must not list multiple repositories. Examine the `dfc.properties` file, ensure that only one (private) connection broker is listed, and only one repository (could be multiple servers serving the same repository) is projected to that connection broker.
2. Run `jmsPackager.sh` under `$DOCUMENTUM_SHARED/jmsTools/bin` to package all the previously deployed WebApps (the WebApps on the original default JMS) and place it under the `$DOCUMENTUM_SHARED/jmsTools/webapps` directory.
3. Run `jmsconfig.sh` under `$DOCUMENTUM_SHARED/jmsTools/bin` to create a new instance of WildFly. In addition to creating a new instance of WildFly, the `jmsconfig.sh` configuration program also deploys the previously packaged WebApps to the new application server.
4. Verify that a new instance of WildFly is created. Start JMS2. For example:

```
% nohup ./startJMS2.sh > nohup.log &
```

Also, run the following IDQL command to confirm if dm_jms_config object is created:

```
select r_object_id, object_name from dm_jms_config
```

If new dm_jms_config object is not created, run the following command to create it:

```
dm_jms_admin.bat -docbase <repository name>
-username <repository owner>
-action add,enableDFC,testDFC,migrate,dumpServerCache,listAll
-jms_host_name <Hostname of JMS> -jms_port <Port of JMS>
-jms_proximity <proximity> -webapps ServerApps
-server_config_id <secondary ID>
```

Enabling JMS HA feature

Content Server 7.3 and later contains changes to the JMS HA feature.

To enable the changed JMS HA feature, set `r_module_name` to **JMS_HA_SETUP_ENABLED** and `r_module_mode` to **1**. To use the unchanged JMS HA feature, set `r_module_mode` to **0**.

For example:

```
append,c,docbaseconfig,r_module_name
JMS_HA_SETUP_ENABLED
append,c,docbaesconfig,r_module_mode
1
save,c,docbaseconfig
```

Then, restart the repository to enable the JMS HA feature.

Note: The *EMC Documentum Content Server 7.2 Installation Guide* contains more information about the unchanged JMS HA features.

JMS HA configurations

Once you enable the JMS HA feature, you should configure the following attributes in the `dm_server_config` object per each Content Server instance. You can also configure using DA.

Server configuration object of each Content Server that maintains the `dm_jms_config` objects list. It represents `jms_mode` for each `dm_jms_config` object. For example:

- `jms_config_id`:


```
jms_config_id [0]: <jms_config_id>
jms_config_id [1]: <jms_config_id1>
jms_config_id [2]: <jms_config_id2>
```
- `jms_type`:


```
jms_type: 1
```

where 1 means that it supports round robin load balancing only and reserved for future release.
- `jms_mode`:


```
jms_mode [0]: 1/2/3
jms_mode [1]: 1/2/3
jms_mode [2]: 1/2/3
```

where `jms_mode` indicates if it has been configured for failover and/or load balancing.

- Load balancing: 1 (0001)
- Failover: 2 (0010)
- Load balancing and Failover: 3 (0011)
- `jms_proximity`: Reserved for future release.

For default configuration, you must use the following command per Content Server instance:

```
apply, c, NULL, REFRESH_JMS_CONFIG_LIST, DEFAULT_JMS_VALUES, B, T
```

where default configuration means Content Server fetches all `dm_jms_config` objects from the repository, adds `dm_jms_config` objects as `jms_config_id` configuration object, and sets the value of the corresponding `jms_mode` to **3** (for both failover and load balancing) for each `jms_config_id`.

Note:

- `dm_server_config` should have at least one `dm_jms_config` object that has either 1 or 3.
- If you want to add additional JMS, then you must configure `jms_config_id` and `jms_mode` values manually in the server configuration object for newly created `dm_jms_config` object. Otherwise, it resets to default values for all Java Method Servers when you run the `REFRESH_JMS_CONFIG_LIST` apply command.

Method location options for JMS HA

- **LOCAL**: Methods with **JMS_LOCATION=LOCAL** means, it must be run on the same host JMS as the originating Content Server.
- **ANY**: Methods with **JMS_LOCATION=ANY** means, it can be run on any JMS as long as it is on same LAN as the originating Content Server.

Note: If you do not configure any **JMS_LOCATION** attribute, Content Server uses the **JMS_LOCATION=ANY** option.

- **REMOTE**: Methods with **JMS_LOCATION=REMOTE** means, it runs on a remote JMS. Remote JMS is associated to a remote Content Server (RCS). If a Content Server needs to run a method with setting as **JMS_LOCATION=REMOTE**, it must send HTTP POST request to its associated remote Java Method Servers.
- **JMS_ID**: Method **JMS_LOCATION=<jms_config_id>** means, it runs on a particular JMS.

Prerequisites

- For methods requiring trusted authentication, you must set `a_extended_properties` to **JMS_LOCATION=LOCAL**. Otherwise, setting these methods results in authentication issues when executed in cross communication between Content Server and JMS.
- After enabling the JMS HA feature in the `docbase_config` object, you must manually run the following command for each Content Server:

```
apply, c, NULL, REFRESH_JMS_CONFIG_LIST, DEFAULT_JMS_VALUES, B, T
```

Restart the repository. Check for `dm_jms_config` entries in `dm_server_config`. The values of `cms_config_id` and `cms_mode` attributes is added.

- In JMS HA, the `a_special_app` attribute does not honor the **Workflow job** keyword. The value of this attribute must be either **workflow** or **blank**.
- In `dm_jms_config`, the value of `cms_name[0]` must be `do_method` and the value of `cms_url[0]` must be the respective URL.
- In a Docker setup, you must enable the following flag:

```
append,c,docbaseconfig,r_module_name
JMS_HA_AUTO_REFRESH_DISABLED
append,c,docbaseconfig,r_module_mode
1
```

Restart the repository. Then, you must manually run the following REFRESH command per Content Server instance whenever the JMS status changes:

```
apply,c,NULL,REFRESH_JMS_CONFIG_LIST
```

- If the system and custom methods need JMS failover, ensure to expose the failover to the method. Set `is_restartable` to **T**. By default, it is **F**.

JMS status

You can run the following commands to:

Refresh the JMS:

```
apply,c,NULL,REFRESH_JMS_CONFIG_LIST
```

Check the status of JMS:

```
apply,c,NULL,DUMP_JMS_CONFIG_LIST
```

Enabling and understanding logs

Perform the following:

- Enable the debug tracing on `C:\Documentum\<WildFly_version_folder>\server\DctmServer_secondJMS\deploy\ServerApps.ear\DmMethods.war\WEB-INF\web.xml` of JMS, set the value of `trace` to **t**.

The trace details are logged in `server.log`.

- Enable `trace_launch` on Content Server. Run the following commands from IAPI:

For HTTP POSTs:

```
apply,c,NULL,SET_OPTIONS,OPTION,S,
trace_http_post,VALUE,B,T
```

For complete launch:

```
apply,c,NULL,SET_OPTIONS,OPTION,S,
trace_complete_launch,VALUE,B,T
```

The trace details are logged in `$Documentum\<WildFly_version_folder>\server\DctmServer_MethodServer\log\server.log`.

- Enable `trace_launch` at the method level while running the method enable trace launch option.

The trace details are logged in a separate log file for each method under `$Documentum\dba\log\xxxxxx\sysadmin\<Method_Name>.txt`

- Installer log files
- JMS configuration program logs:
 - `C:\Documentum\jmsTools\bin\` or `$DOCUMENTUM_SHARED\jmsTools\bin/`
 - `install.log`
 - `setupError.log`
 - `C:\Documentum\dba\config\docase\dm_jms_config_setup.out`

Error messages

Error message	Description
DM_METHOD_E_HTTP_COMMUNICATION	Application server to which the Content Server wants to send the POST request to, is unavailable or down.
DM_METHOD_E_HTTP_POST_APP_SERVER_NAME_NOTFOUND	No matching <code>app_server_name</code> or <code>servlet_name</code> value found in any of the <code>dm_server_config</code> or <code>dm_jms_config</code> objects.
DM_METHOD_E_NO_JMS_AVAILABLE2	Content Server understands that all the Java Method Servers are currently down and do not accept any POST requests.
DM_METHOD_E_HTTP_POST_FAILED	Content server could not send the POST request to the application server.

Supported HA configurations

JMS supports the following HA configurations:

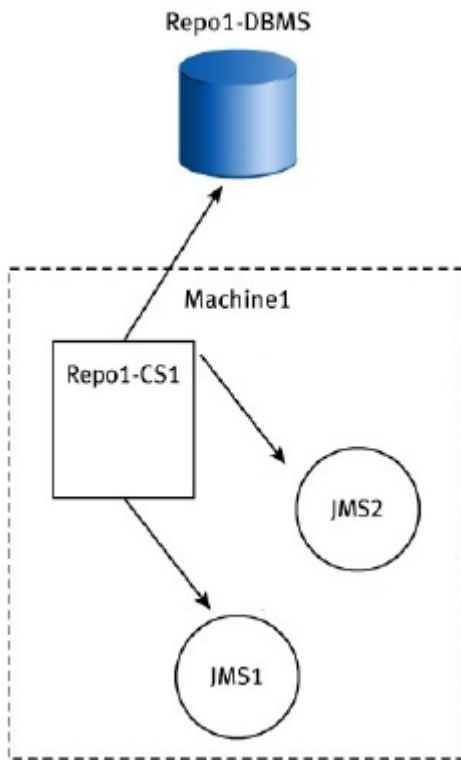
- Content Server and Java Method Servers on a single host
- Content Server and Java Method Server on two hosts
- Content Server and Java Method Servers on multiple hosts

Content Server HA deployment involves two or more Content Servers. Java Method Server High-Availability involves adding Java Method Servers to additional Content Servers such that each Content Server has a dedicated Java Method Server. Java Method Server High-Availability is automatically enabled by associating each Content Server with their dedicated Java Method Server.

Content Server and two or more Java Method Servers on a single host

The illustration depicts two Content Servers and their embedded instances of JMS set up on a single machine, serving one repository.

Figure 6. Content Server and two or more Java Method Servers on a single host



When installing the first Content Server, JMS is installed by default. When adding a Content Server on the same host, the JMS for the second Content Server is not installed. Instead, it shares the JMS of the first Content Server.

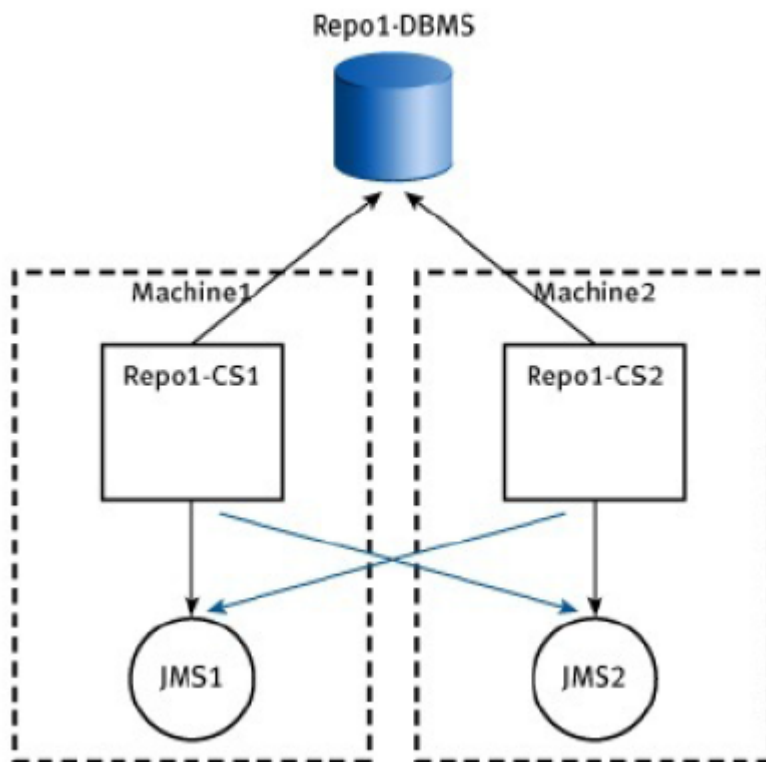
To install and configure the main Content Server

1. Install the main Content Server.
2. After the installation completes, run the Content Server configuration program. The Content Server Configuration program guides you through creating a repository.
The installation of the main Content Server also installs an instance of JMS. A single `dm_jms_config` object for that instance of JMS is created in the repository.
3. Add the additional Java Method Servers. [Installation of additional JMS, page 154](#) contains the instructions.

JMS for HA on multiple hosts

The illustration depicts two embedded JMS instances, each connected to its own Content Server on two hosts supporting one repository to two JMS instances set up for HA with Content Servers on two hosts supporting one repository.

Figure 7. JMS for HA on multiple hosts



Each connected to its own Content Server on multiple hosts supporting one repository to multiple JMS instances set up for HA with Content Servers on multiple hosts supporting one repository.

To install and configure the main Content Server

1. Install the main Content Server.
2. After the installation completes, run the Content Server configuration program. The Content Server Configuration program guides you through creating a repository.
The installation of the main Content Server also installs an instance of JMS.

To add Content Servers

1. On multiple hosts, install the main Content Server.
2. On Windows, run `$DM_HOME\install\cfsConfigurationProgram.exe` as the installation owner; on UNIX and Linux, run `$DM_HOME/install/cfsConfigurationProgram.bin` as the installation owner.
3. Provide the appropriate information when prompted to do so by the program.

4. When you are prompted for the service name, EMC recommends that you choose a service name that is different from that of the primary Content Server.

To configure JMS for HA

1. Enable the `JMS_HA_SETUP_ENABLED` flag.
2. Add the JMS configuration IDs or set the `jms_mode` attribute for each Content Server's server configuration object.
Or
Run the following command for each Content Server instance to configure the defaults values:
`apply,c,NULL,REFRESH_JMS_CONFIG_LIST,DEFAULT_JMS_VALUES,B,T`
3. Set `a_extended_properties` to **JMS_LOCATION=LOCAL** for all methods requiring trusted authentication.
4. In JMS HA, the `a_special_app` attribute does not honor the **Workflow job** keyword. The value of this attribute must be either **workflow** (if BPM is installed) or **blank**.
5. Ensure that in `dm_jms_config`, the value of `servlet_name[0]` is `do_method` and the value of `base_url[0]` is the respective URL.
6. Restart the repository.

Two or more Content Servers and two or more Java Method Servers on a multiples host

You can install and configure mixed mode Content Server and Java Method Server combinations.

[Content Server and two or more Java Method Servers on a single host, page 159](#) and [JMS for HA on multiple hosts, page 160](#) sections contains the instructions.

Installing Content Server with Microsoft Cluster Services

This section describes how to install and configure Content Server to provide failover support under Microsoft Cluster Services.

Microsoft Cluster Services overview

Microsoft Cluster Services supports the Active/passive clusters and Active/active forms of clustering.

In a cluster environment, every service that the cluster runs uses resources of the cluster node. Every service has its own resources, such as hard drive, IP address, and network name, assigned to it. All resources that a clustered service uses form a resource group. The connection broker and Content Server are a part of this resource group. In a cluster, all resources form a virtual server that can move from one physical server to another to provide failover support.

Note: Lockbox needs to be configured on Content Server depending on how Microsoft Cluster Services is setup.

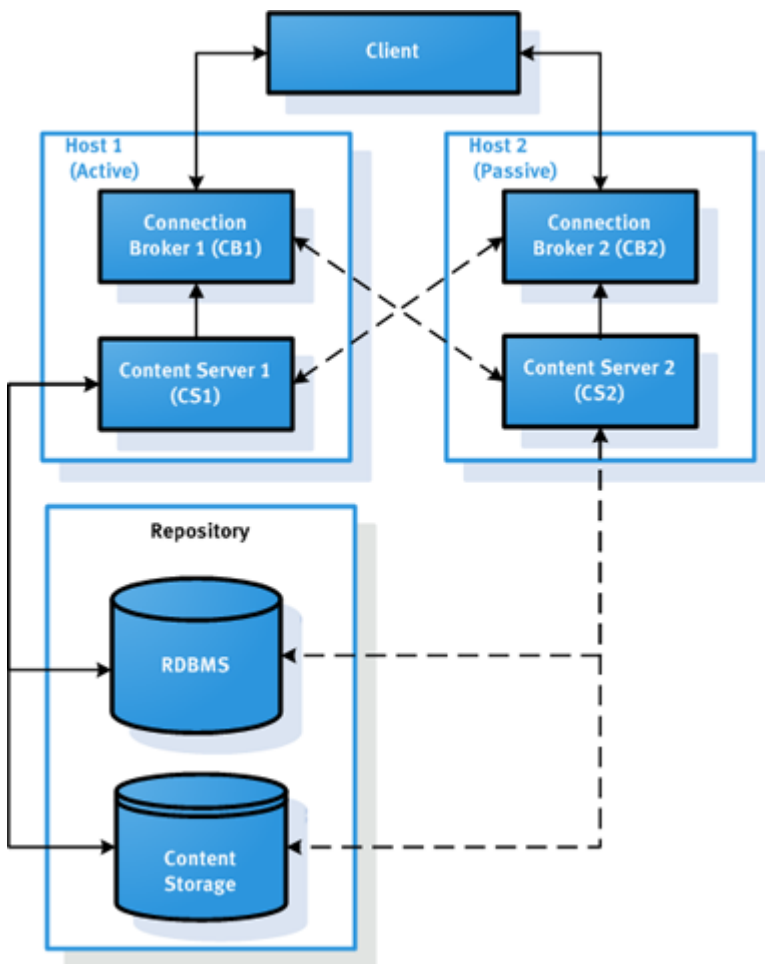
Choosing a configuration

A Content Server installation supports two types of cluster service configurations:

- active/passive
- active/active

This section provides detailed installation instructions for both configurations. Choose the configuration based on available hardware and your organization's business needs. The figure illustrates Content Server and connection broker setup in an active/passive cluster.

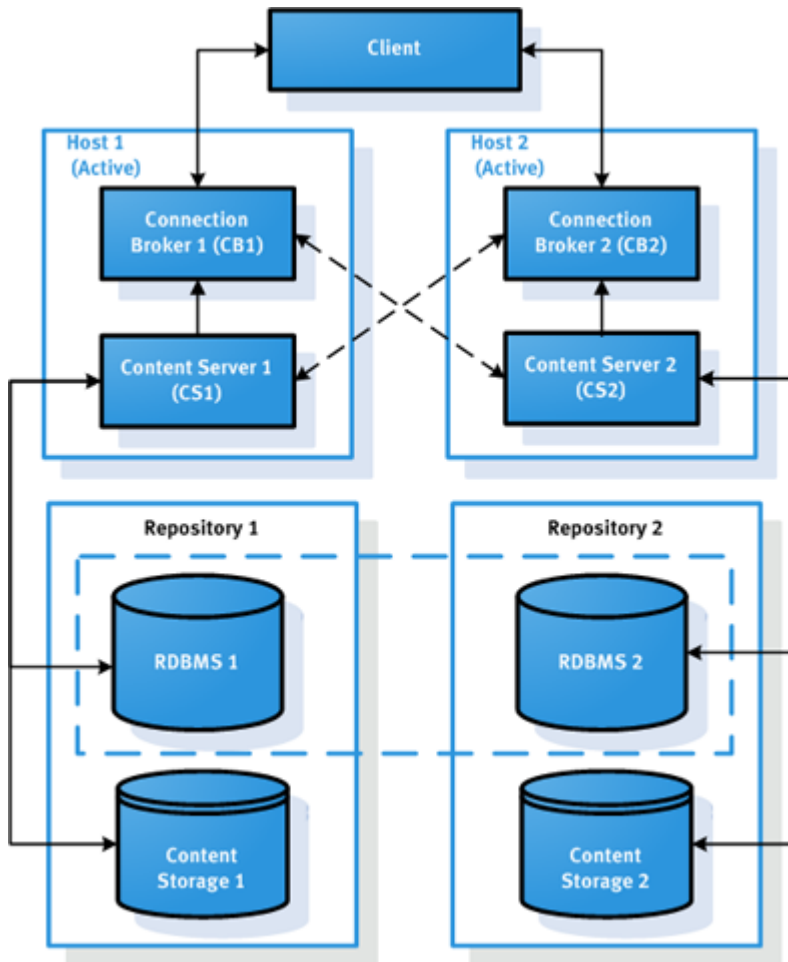
Figure 8. Active/passive cluster



In an active/passive configuration, both nodes support the same repository. Content Server and connection broker run on the primary node. If the primary node fails, the secondary node that was on standby takes ownership of the resource group. After the cluster resources are brought online on the secondary node, the connection broker and Content Server start on the secondary mode.

In an active/active configuration, each node supports a different repository. Each node is considered the standby to the other node. Each node owns its own resource group. Each resource group has its own virtual IP address, a virtual host name, shared disk, connection broker, and Content Server. The figure illustrates Content Server and connection broker setup in an active/active cluster.

Figure 9. Active/active cluster



If a node fails or is taken offline, its resource group is moved to the remaining node in the cluster. The remaining node then manages two resource groups. When the failed node is running again, the cluster administrator can move one resource group back.

If you switch the content transfer from one node to the other during a content transfer operation, it fails. This occurs because ACS may not be available on the second node to perform the content transfer. If a subsequent content transfer operation is done, the content transfer takes place through the Application Server (for example, Webtop) and is expected to succeed. If there are issues in transferring large files, configuration changes may be required.

Pre-installation requirements

Before you install and configure Content Server and a repository under Microsoft Cluster Services, perform the preinstallation tasks. [Pre-installation requirements and tasks, page 89](#) contains the information.

Whether you are configuring an active/passive cluster or an active/active cluster, set up the shared disks to be used by the repositories. Ensure that the shared disks include a directory to use for content storage.

If you are configuring an active/active cluster, the user who configures Microsoft Cluster Services must have read and write permissions on both nodes on the directories where the connection broker logs reside.

Configuring an active/passive cluster

Follow these procedures to configure an active/passive cluster and install servers and a Content Server.

Note: When setting up an active/passive cluster in a Windows cluster environment, configuration of a repository on the second cluster node can fail if you are using different `aek.key` files and `dbpassword.txt` files for each node. In active/passive cluster configuration, both the primary and secondary node must have the same `aek.key` file (located in `\Documentum\dba\secure`) and `dbpasswd.txt` (located in `\Documentum\dba\config\docbase_name`). When creating the environment manually copy the two files from the primary node to the secondary node.

Creating the cluster resource group

Use this procedure to set up the cluster resource group.

To create the cluster resource group:

1. Use the MSCS Cluster Administrator utility to create a cluster resource group that contains the following resources:
 - A virtual IP address
Content Server uses this IP address. Other products can share the IP address because the connection broker and Content Server only listen on particular ports.
 - A virtual network host name
The virtual network host name is for the virtual IP address.
 - A shared disk partition
This is the location of the repository data directory, where content files are located.
2. Move the resource group between the primary and standby hosts to ensure that the virtual IP address, virtual network host name, and shared disk partition fail over properly.
 - a. In the Cluster Administrator, right-click the cluster group name.

- b. Click **Move Group**.
The resource group is toggled between the hosts and the owner name changes.
3. Ensure that the first node owns the resource cluster group.

Installing Content Server software on the nodes

The first part of the installation process copies files from the installation media to the hard disk. On each of the two nodes, copy the files from the installation media to create a Content Server installation on each node. Use the same drive letter on each node for the installation.

Configuring Content Server

The second part of the installation involves configuring Content Server and its associated connection broker, which includes configuring them for Microsoft Cluster Services.

To configure Content Server on Node1:

1. Move the cluster control to Node1.
2. Extract the installation files, launch `serverSetup.exe` and install Content Server on Node1.
3. Configure the connection broker and repository on Node1.

Note: Specify the location of the Data folder which will be the location of your file stores. It is recommended that this folder be on a shared drive on the Microsoft Cluster setup or on a shared drive accessible to both the nodes.

To configure Content Server on Node2:

1. Turn off the connection broker and repository services on Node1 and move the cluster control to Node2.
2. On Node2, follow the steps used on Node1 using the same **Repository name** and **Repository ID** as in Node1.
3. In the **Select Database and Server Account** screen, choose **Use existing SQLServer user account and database** to use the database created earlier.
4. Installer prompts you to enter the details of the existing database. Type the details.
5. Type the information of the SMTP server and the repository owner's email address. Type the same details which were provided in Node1.
6. Choose the same optional modules selected on Node1.
7. If you have selected the RSA Key Manager option in the previous menu, then Installer prompts you for the RKM details in the next few screens. Type the same details used earlier.
8. Installer proceeds with the connection broker configuration and an error message is shown due to failure to connect to the repository. At this stage stop the installation.

9. Now turn off all the connection broker and repository services on Node2. Copy the `$DOCUMENTUM\dba\secure\sek.key` and `$DOCUMENTUM\dba\config\repository_name\dbpasswd.txt` files from Node1 to corresponding locations on Node2.
10. If you have configured the repository with RKM then you also need to copy the files `rkm_config.ini`, `client_reg_rkm.cfg`, `rkm_svc.cfg`, `rkmcachepasswd.txt`, `rkmclientcert.pl2`, `rkmpasswd.txt` and `rkmrootcert.pem` from `$DOCUMENTUM\dba\config\repository_name` on Node1 to `$DOCUMENTUM\dba\config\repository_name` on Node2. Restart the repository on the Node2 and check the logs to ensure it has started successfully.
11. To configure more nodes, move the cluster control to that node and follow steps on Node2.

Note: If lockbox is enabled, copy all the lockbox related files from Node1 to Node2 and then run the `dm_crypto_boot` and `dm_crypto_create` commands to enable lockbox work on Node2.

The setup of connection broker and repository configuration on two nodes required for the purpose of using Microsoft Cluster Services for failover is complete.

Configuring the connection brokers

You need to configure the connection brokers on both nodes to listen on the virtual network host.

To configure connection brokers:

1. Click **Documentum > Documentum Server Manager** to start Server Manager.
2. Click **Docbroker**, then click **Edit Service**.
3. Add the following line to the service command:
`-host virtual_network_host_name`
For example, if the virtual network host name is `dmcluster1`:
`-host dmcluster1`
Click **OK**.
4. Highlight **repository** and click **Edit server.ini**.
5. Edit the `DOCBROKER_PROJECTION_TARGET` section of the `server.ini` file:

```
[DOCBROKER_PROJECTION_TARGET]
host=virtual_network_host_name
```
6. Save the `server.ini` file.
7. Navigate to `C:\Documentum\config` and open the `dfc.properties` file in a text editor.
8. Edit the primary connection broker entry of the `dfc.properties` file:
For example:
`dfc.docbroker.host[0] = dmcluster1`
9. Save the `dfc.properties` file.
10. Repeat this procedure on the other node.

Creating additional cluster resources on Microsoft Cluster Services

Use this procedure to create cluster resources for the connection broker and Content Server. Perform the procedure only on the node that presently owns the existing resource group.

To configure the connection broker and repository for failover:

1. Navigate to **Administrative Tools > Failover Cluster Management**.
2. Right-click the services node and choose the service you created.
3. Choose **Generic Service** to configure for high availability and click **Next**.
4. Choose the **Documentum Docbroker Service** connection broker service from the list, proceed with the wizard and complete configuring the connection broker and repository for failover.
5. Right-click the new service, and choose **Properties**.
6. In the General tab, choose **NODE1** and **NODE2** in the Preferred owners list. Click **OK**.
7. Right-click the **Documentum Docbroker Service** connection broker service, and choose **Properties**.
8. In the **General** tab, ensure that the **Use Network Name for computer name** option is selected by default, and click **OK**.
9. Right-click the **Documentum Docbroker Service** connection broker service in the console, and choose the **Bring this resource online** option. The resource is brought online and the Online icon is displayed in the Status column of the Summary screen.
10. Right-click the configured service, choose **Add a resource > Generic Service**.
11. Choose **Documentum Docbase Service** in the Select Service list of the Select Service screen.
12. Click **Next** and click **Next** again.
13. Click **Finish**. The Documentum Docbase Service is added to the Other Resources list.
14. Right-click the **Documentum Docbase Service** in the Summary screen, and choose Properties.
15. In the **Dependencies** tab, add the Documentum Docbroker Service.
16. In the **General** tab, ensure that the Use Network Name for computer name option is selected, and click **OK**.
17. Right-click the **Documentum Docbase Service** in the Summary screen console, and choose the **Bring this resource online** option. The resource is brought online and the Online icon is displayed in the Status column of the Summary screen.
18. Right-click the configured service, choose **Add a resource > Generic Service**.
19. Choose **Documentum Java Method Server** in the Select Service list of the Select Service screen.
20. Click **Next** and click **Next** again.
21. Click **Finish**. The Documentum Java Method Server is added to the Other Resources list.
22. Right-click the **Documentum Java Method Server** in the Summary screen, and choose **Properties**.
23. In the **Dependencies** tab, add the Documentum Docbroker Service and Documentum Docbase Service.

24. In the **General** tab, ensure that the Use Network Name for computer name option is selected, and click **OK**.
25. Right-click the **Documentum Java Method Server** in the Summary screen console, and select the **Bring this resource online** option. The resource is brought online and the Online icon is displayed in the Status column of the Summary screen.

Verifying failover

After you complete the preceding procedures, verify that failover works properly.

Note: Ensure to run the `dm_crypto_boot` utility on all the Content Server nodes only if it has the custom passphrase before switching Content Server services to any node.

To verify failover:

1. On a client computer, ensure that the `dfc.properties` entries refer to the virtual network host name or virtual IP address.
2. Connect to the repository from the client computer.
3. Start the Cluster Administrator utility.
4. Move the resource group from the node where it is running to the other node.
5. After the resource group comes online on the other node, verify that the client can run queries.

Configuring an active/active cluster

In an active/active cluster, each node initially has its own repository and Content Server. You create two cluster resource groups, and each node owns one. If a Content Server fails on one node, a second Content Server starts on the second node to keep the repository on the first node running.

Each cluster resource group has the following:

- virtual IP address
- virtual network host name
- one shared disk drive (where the repository resides)
- one connection broker
- one Content Server

Creating the first cluster resource group

Create the first cluster resource group. [Creating the cluster resource group, page 164](#) contains the information.

Installing Content Server software on the hosts

On each of the two nodes, copy the files from the installation media to create a Content Server installation on each node.

Note: Use the same drive letter on each node.

Configuring Content Server on the first and second nodes

In an active/active configuration, configure a Content Server on each of the two nodes. The procedure for configuring Content Server in an active/active cluster is the same as configuring Content Server in an active/passive cluster. Follow the instructions on the first and the second node in the cluster.

Configuring the second cluster resource group

Create the second cluster resource group.

Modifying server.ini and dfc.properties

You might want to edit `server.ini` and `dfc.properties` on both nodes to ensure that each repository projects to the connection brokers on the two nodes.

In the following examples, assume that the virtual network hosts for the two cluster resource groups are called `dmcluster1` and `dmcluster2`. It does not matter which cluster resource group is primary and which is backup.

Edit all four of the `server.ini` files so that they read as follows:

```
[DOCBROKER_PROJECTION_TARGET]
host=dmcluster1
[DOCBROKER_PROJECTION_TARGET_1]
host=dmcluster2
```

Edit the two `dfc.properties` files so that they read as follows:

```
dfc.docbroker.host[0]=dmcluster1
dfc.docbroker.host[1]=dmcluster2
```

Configuring the connection broker and repository for failover

The procedure for configuring the connection broker and repository for failover in an active/active cluster is the same as configuring them in an active/passive cluster. But the configuration for failover needs to be done separately for each of the two repository services binding them with their respective dependant resources.

Verifying failover

After you complete the preceding procedures, verify that failover works properly.

To verify failover:

1. On a client computer, ensure that the `dfc.properties` entries refer to both virtual network host name or virtual IP address.
2. Connect to both repositories from the client computer.
3. Start the Cluster Administrator utility.
4. Move the two resource groups back and forth between the nodes.
5. After a resource group comes online on a new node, verify that the client can run queries.

Uninstalling Content Server

This section explains how to delete a repository or connection broker and how to uninstall an existing Content Server. Do not uninstall an existing installation to upgrade to a new Content Server release, because all upgrades are based on an existing installation. Use the procedures in this section *only* if you want to uninstall an existing Content Server, a repository and its contents, a connection broker, or a Content Server software installation.

To delete a repository or connection broker or uninstall Content Server, you need to meet the following requirements:

- Be able to log in as the installation owner
- Have sufficient database privileges to drop tablespaces or databases

Uninstalling components

Uninstall the software components in this order:

1. Stop the Java Method Server service.
2. Delete the repository.
3. Delete any connection broker located on the current host.
4. Uninstall the Content Server software.
5. Uninstall the Index Agent configuration program if full-text indexing is installed.

Deleting a repository

To delete a repository you need to meet the following requirements:

- Be able to log in as the installation owner
- Have sufficient RDBMS privileges to drop tablespaces or databases.

Note: If the repository has a Content Transformation Services (CTS) product installed on it, you need to uninstall the CTS product before deleting the repository. If you do not, the CTS product will not be available in all other repositories.

Log in to the host as the Content Server installation owner, start the Content Server configuration program, and choose **Delete an existing repository** to delete a repository.

Deleting a connection broker

Log in to the host as the installation owner, start the Content Server configuration program, and choose **Delete an a connection broker** to delete a connection broker.

Uninstalling the Content Server software

Use these instructions to uninstall the Content Server software from a host. You can only uninstall the software after deleting all repositories and connection brokers in the installation.

1. Delete all repositories and connection brokers in the installation.
2. Run the Content Server uninstallation program.
 - On Windows, run `$DOCUMENTUM/uninstall/server/uninstall.exe`
 - On UNIX and Linux, run `$DOCUMENTUM/uninstall/server/uninstall`

Optionally, run `$DOCUMENTUM/uninstall/dfc/uninstall` to uninstall the DFC Runtime Environment.

Note: Do not uninstall the DFC Runtime Environment if any other EMC Documentum software is installed on the host.

Troubleshooting

This section contains the information for troubleshooting common Content Server installation problems.

Identifying the problem and resolution

When experiencing a problem, perform the following preliminary actions:

- Ensure that you are connected as the installation owner.
- On UNIX and Linux, ensure that the environment variables are set correctly in the installation owner's environment.
- Review the Content Server installation logs.

Symptom	Cause	Fix
Content Server installation or upgrade fails	You are trying to install or upgrade the Content Server but you are not connected as the installation owner.	Connect using the installation owner account.
<p>While installing a repository, you see an error message that indicates that the user is not a valid UNIX user:</p> <pre>Configuration of the docbase fails with the message 'user must be a valid UNIX user' exec(): 0509-036 Cannot load program /u01/app/documentum /product/5.2/bin /dmisvaliduser because of the following errors: 0509-150 Dependent module libldap50.so could not be loaded. 0509-022 Cannot load module libldap50.so. 0509-026 System error: A file or directory in the path name does not exist.</pre>	<p>Three possible causes:</p> <ol style="list-style-type: none"> 1. The installation owner account does not have the installation owner group designated as the user's primary group. Group ownership of the Documentum binaries is incorrect. 2. The shared library path environment variable is not set correctly. 3. On DB2, the \$DB2_BASE environment variable is not set correctly. 	<p>To fix:</p> <ol style="list-style-type: none"> 1. Make the installation owner group the installation owner's primary group. 2. Set the shared library path environment variable correctly. Setting the required environment variables, page 103 contains the information. 3. General database requirements, page 89 contains the information.
Content Server upgrade appears unresponsive.	There might be a cyclic group.	contains the information.

Symptom	Cause	Fix
<p>On Windows hosts, you see the following error during installation:</p> <pre>Could not initialize interface awt exception ExceptionInitialization Error</pre>	<p>The correct video driver for the video card is not installed on the host.</p>	<p>Review the hardware and software configuration of the host.</p>
<p>You see the following error during an upgrade of an older repository:</p> <pre>Failed to retrieve serverconfig object with name <serverconfigname>. ***Failed to encrypt passwords for docbase ec_epac, status -1057226550 **Operation failed ** [DM_CRYPT0_E_NO _LOCAL_COMPONENT_STORE] error: "No local component store for server" Please read error log C:\WINNT \Temp\dm_chec_bin .ServerConfigurator.log for more information.</pre>	<p>The dm_ContentReplication method has some parameter arguments left over from EDMS98.</p>	<p>Delete the following entry from the dm_ContentReplication method:</p> <pre>serverconfigname [domain\]user,passwd</pre>

Symptom	Cause	Fix
<p>You see the following errors during upgrade from 5.3 SP6 on Oracle 10:</p> <pre>Tue Feb 22 21:48:08 2005 098000 [DM_SESSION_I_INIT_BEGIN]info: "Initialize dmContent." Tue Feb 22 21:48:08 2005 567000 [DM_SESSION_I_INIT_BEGIN]info: "Initialize dmiSubContent." Tue Feb 22 21:48:08 2005 598000 [DM_TYPE_MGR_E_CANT_FIND_TABLE]error: "Failure to find table dmi_subcontent_sv as part of fetch of type dmi_subcontent: error from database system is ORA-24372: invalid object for describe" Tue Feb 22 21:48:08 2005 598000 [DM_SESSION_E_INIT_FAILURE1]error: "Failure to complete dmiSubContent initialization."</pre> <p>You might also see this message:</p> <pre>ORA-24372: invalid object for describe</pre>	<p>Invalid Oracle views belonging to types <code>_sv</code>, <code>_sp</code>, <code>_rv</code>, and <code>_rp</code>.</p> <p>A view in Oracle becomes invalid when the base tables it references change (for example, by adding/dropping a column, or dropping a unique constraint index).</p>	<p>Make the views valid before upgrading Content Server.</p> <p>To determine which views in the Oracle installation are invalid, you can run the following query from SQLPLUS logging in as the repository owner:</p> <pre>select object_name, object_type from user_objects where status='INVALID';</pre> <p>To recompile the views:</p> <pre>ALTER VIEW view_name COMPILE;</pre> <p>The Oracle-supplied package named <code>DBMS_UTILITY</code> has a procedure named <code>COMPILE_SCHEMA</code>. This procedure will compile all stored code, views, and so on, for the schema provided. The best way to compile all database objects that are invalid is to use a script in the <code>\$ORACLE_HOME/rdbms/admin</code> directory named <code>utlrp.sql</code>. This script finds all objects in the data dictionary that are invalid and compiles them. This script is typically mentioned in patch notes but you can use it any time a schema change occurs.</p>
<p>On AIX, you see the following exception:</p> <pre>Needed JSSE provider IBMJSSE Cannot be found</pre>	<p>IBM JDK 1.7 support is only from AxM Server 6.2.</p>	<p>Use the runtime jar <code>axm-runtime-api-6.2.jar</code> to connect to AxM Server 6.2.</p>

Symptom	Cause	Fix
Unable to start the repository.	Changing the default passphrase to a custom passphrase.	During Content Server installation, a keystore is created that contains a passphrase that is used for encryption. After installation, you can change the default passphrase to a custom passphrase. If you create a custom passphrase after Content Server installation, any time you restart the server host you need to run the <code>dm_crypto_boot</code> utility. <i>EMC Documentum Content Server Administration and Configuration Guide</i> contains the instructions and details on encryption, keystores, and passphrases.
Error while configuring RCS.	Changing the fingerprint of the machine.	Run the <code>dm_crypto_manage_lockbox</code> utility with the <code>-resetfingerprint</code> option and then restart the machine. <i>EMC Documentum Content Server Administration and Configuration Guide</i> contains the details on the utility.

While installing on Linux, Installer hangs when the number of mount points exceeds 4000

On a Linux machine, installation can hang when there are too many mount points open in the box. Check the output in bytes by entering the following command:

```
mount | wc -c
```

Installer hangs when the number returned by the previous command exceeds 4000.

Workaround: Copy and paste the following into your Telnet session and run the installation as usual:

```
cd /tmp
mkdir bin.$$
cd bin.$$
cat > mount <<EOF
#!/bin/sh
exec /bin/true
EOF
chmod 755 mount
export PATH=`pwd`: $PATH
```

After the installation completes, delete /tmp/bin.\$\$ by running the following command (in the same Telnet session):

```
rm -r /tmp/bin.$$
```

Recovering from a failed repository configuration or upgrade

If repository configuration fails, whether you are upgrading an existing repository or creating a new one, you can recover from the failure.

Typical reasons for a failure include problems with the database connection or errors in Content Server creation. Before you proceed with the following instructions, read the Content Server installation logs and correct any problems.

To recover from a failed installation or upgrade:

1. Correct any problems noted in the Content Server installation logs.
2. Restart the Content Server configuration program.
3. Select **Repository > Upgrade an existing repository**.
4. Select the repository where the failure occurred.
5. Check **Upgrade**.

This takes you through the configuration steps again and reruns the scripts that create the repository.

Recovering from a stalled Content Server upgrade

A Content Server upgrade that stalls in the middle or takes hours to complete can be caused by *cyclic groups*. A cyclic group is a subgroup of a member group, causing the Content Server to cycle during the upgrade. If the Content Server has encountered a cyclic group, the last line of the Content Server log is:

```
Thu Jun 28 14:00:14 2007 715540  
[DM_SESSION_I_INIT_BEGIN]info:"Initialize dmGroup."
```

Use the following instructions to identify the cyclic group. After you locate the cyclic group, contact EMC Documentum Technical Support for assistance in correcting the problem, which requires direct SQL Server statements in the database.

To identify and correct a cyclic group:

1. From the operating system, stop the Content Server startup.
 - On Windows, open Task Manager, select the correct Content Server process on the Processes tab, and click **End Process**.
 - On Linux, determine the correct Content Server process and use the kill command to end the process.

2. If you are on Linux, restart the Content Server using the `-osqltrace` option:
`dm_start_repositoryname -osqltrace`
3. If you are on Windows, edit the Content Server startup command, then restart the Content Server.
 - a. Click **Documentum Server Manager**.
 - b. Choose the correct repository.
 - c. Click **Edit Service**.
 - d. In the **Command** field, add `-osqltrace` after the repository name. Click **OK**.
 - e. Restart the Content Server.
4. When the Content Server appears to be unresponsive, open the Content Server log and identify the query that is looping.
 If there is a cyclic group, the last query in the log is recorded multiple times and takes this format:

```
Thu Jun 28 13:33:17 2007 435439: 21547[1]
SELECT SB_.R_OBJECT_ID FROM repository_owner.dm_group_s SB_
WHERE (SB_.R_OBJECT_ID=:objectp AND SB_.I_VSTAMP=:versionp)
Thu Jun 28 13:33:17 2007 435608: 21547[1] :objectp = 1200fb8080000909
Thu Jun 28 13:33:17 2007 435608: 21547[1] :versionp = 0
```

 In the preceding example, the cyclic group has the `r_object_id` of 1200fb8080000909.
5. Run the following query:

```
SELECT group_name
FROM dm_group_s
WHERE r_object_id='r_object_id_of_cyclic_group'
```

 This query returns the name of the group, which you need for determining which group is the cyclic group.
6. Run the following query:

```
SELECT groups_names
FROM dm_group_r
WHERE r_object_id = 'r_object_id_of_cyclic_group'
```

 The query returns the names of each group that is a member of the problem group.
7. For each of the group names returned, run this query:

```
SELECT r_object_id from dm_group_s
where group_name = 'member_group_name'
```

 The query returns the `r_object_id` for each member group.
8. Repeat steps 6 and 7 iteratively for each subgroup until you locate the cyclic group.
9. Contact EMC Documentum Technical Support for assistance in correcting the problem.

Identifying issues for Certificate-based SSL communication

This section lists some of the common issues that occur during configuration of Certificate-based SSL communication, causes, and its resolutions.

Connection broker startup fails

1. Check if the connection broker keystore and keystore password files are available in the \$DOCUMENTUM/dba/secure directory.
2. Check if an entry for the connection broker keystore and keystore password files is available in the connection broker configuration file (docbroker.ini).

- Ensure to add the following entry in the docbroker.ini file:

```
[DOCBROKER_CONFIGURATION]
secure_connect_mode=secure
keystore_file=broker.pl2
keystore_pwd_file=broker.pwd
cipherlist=AES128-SHA
crypto_keyname = <CSaek>
crypto_lockbox = <crypto_lockbox>
```

- crypto_keyname = <CSaek> where <CSaek> is the AEK key name that is used to encrypt the password in the broker.pwd file. Use the following syntax:

```
dm_encrypt_password -encrypt password -file broker.pwd -keyname <CSaek>
[-passphrase <AEKPassphrase>]
```

- crypto_lockbox = <crypto_lockbox>: It is used if lockbox is present. When running dm_encrypt_password, use the following syntax:

```
dm_encrypt_password -encrypt password -file broker.pwd -keyname <CSaek>
[-passphrase <AEKPassphrase>] -lockbox <lockbox>
-lockboxpassphrase <lockboxpassphrase>
```

3. Check if the format of the connection broker keystore is PKCS #12.

The following commands list the keys in the keystore if the keystore is in the PKCS #12 format.

Using OpenSSL:

```
openssl pkcs12 -info -in <keystore>
```

Using Keytool:

```
keytool -list -v -storetype pkcs12 -keystore <keystore>
```

4. Check if the password in the keystore password file is correct.

Note: Specify the password in the plain text (without encryption) format, to perform the test.

Server startup fails

1. Check if the server keystore, server keystore password, and server trust store files are available in the \$DOCUMENTUM/dba/secure directory.
2. Check if the entry for the server keystore, server keystore password, and server trust store files is available in the server configuration file (server.ini).
3. Check if the format of the server keystore is PKCS #12.
4. Check if the server trust store is in the PKCS #7 binary (der) format. For verification, check if the following command dumps it successfully:

```
openssl pkcs7 -in <Keystore> -inform der
```

Example:

```
openssl pkcs7 -in server-trust.p7b -inform der
```

Server not able to connect to connection broker

1. Check whether the connection broker keystore contains the correct key and public certificate.

Use the following command to print the keys in keystore:

```
openssl pkcs12 -info -in <keystore>
```

Using Keytool:

```
keytool -list -storetype pkcs12 -keystore <keystore>
```

2. Check whether the connection broker is sending proper certificates.

The following command starts a simple client that connects to the SSL Server and displays the certificate chain sent by the Server:

```
openssl s_client -showcerts -debug -connect  
<SSL_Server_IP>:<SSL_Server_Port>
```

Example:

```
openssl s_client -showcerts -debug -connect 10.8.53.24:1490
```

3. Check if the server trust store contains the connection broker's public certificate or CA certificate chain used to sign connection broker's public certificate.

Use the following command to display all certificates in the trust-store:

```
openssl pkcs7 -in <trust-store> -inform der -print_certs -text
```

4. Check if the server's trust store contains the connection broker's public certificate or CA certificate chain used to sign connection broker's public certificate.

Use the following command to dump the trust store:

```
openssl pkcs7 -in <trust-store> -inform der -print_certs -text > CA.PEM
```

Use the following command to check connectivity using this trust store:

```
openssl s_client -showcerts -debug -Cafile CA.pem -connect  
<SSL_Server_IP>:<SSL_Server_IP>
```

Example:

```
openssl s_client -showcerts -Cafile CA.pem -debug -connect 10.8.53.24:1490
```

If there are no problems with the trust store, the verification will return (ok).

Clients are unable to connect to the connection broker

1. Check if proper entries are available in the `dfc.properties` file and if the trust store file is available.
2. Check if the DFC trust store contains the connection broker's public certificate or CA certificate used to sign connection broker's public certificate.
3. Run the following command to dump the trust store contents (do not specify the storetype as the default type is JKS):

```
keytool -list -keystore <keystore> -storepass <storepass>
```

Clients are unable to connect to the Server

1. Check if proper entries are available in the `dfc.properties` file and if the trust store file is available.
2. Check if the DFC trust store contains the server's public certificate or CA certificate used to sign the server's public certificate.
3. Run the following command to dump the content of the trust store (do not specify the store type as the default type is JKS):

```
keytool -list -keystore <keystore> -storepass <storepass>
```
4. Check if the correct value has been specified for the trust store password in the `dfc.properties` file. For verification, specify the password in plain text.

Additional information

Run the following command to verify whether the connection is secure and if the encryption algorithm used while starting the client is correct:

```
jvm parameter -Djavax.net.debug=all
```

Content Server error messages and causes

Error message	Cause
<p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_init(). Error (705) SSL error(10100) in call to SSL_CTX_new(). Winsock error: 10100."</p> <p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_open(). Error (501) Network Library not initialized. Extended network error: 0."</p>	<ul style="list-style-type: none"> • Server keystore is missing • Format of the Server keystore is different from PKCS #12 • Server keystore is corrupt • Server keystore password file contains the wrong password • Server keystore password file is missing
<p>[ERROR] [AGENTEXEC 3088] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,"",try_native_first, status: 0, with error message [DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure</p>	<p>Server keystore is empty</p>

<p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_init(). Error Unknown error code 10008 (_nl_error_ = 0). Extended network error: 0."</p> <p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_open(). Error (501) Network Library not initialized. Extended network error: 0."</p>	<ul style="list-style-type: none"> • Server trust store is missing • Server trust store is corrupt
<p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_init(). Error Unknown error code 1795 (_nl_error_ = 0). Extended network error: 0."</p> <p>[DM_SERVER_E_START_NETWORKWISE]error: "The server failed to start due to an error returned by the Netwise networking subsystem. Failed API: nl_open(). Error (501) Network Library not initialized. Extended network error: 0."</p>	<p>Server trust store is in the wrong format</p>
<p>[DM_DOCBROKER_E_NETWORK_ERROR]error: "An error occurred performing a network operation: (Unknown error code 112 (_nl_error_ = 0)). Network specific error: (Extended network error: 0)."</p> <p>[DM_DOCBROKER_E_CONNECT_FAILED_EX]error: "Unable to connect to DocBroker. Clients please check your dfc.properties file for a correct host. Server please check your server.ini or target attributes in server config. Network address: (INET_ADDR: family: 2, port: <port>, host: <host_name> (<host_ip>, 1835080a))."</p>	<ul style="list-style-type: none"> • Server trust store is empty • Connection broker keystore is empty

<p>[ERROR] [AGENTEXEC 2692] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,'"try_native_first, status: 0, with error message [DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "<docbroker_name>:<port>" failed</p> <p>[DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>javax.net.ssl.SSLException:</p> <p>java.lang.RuntimeException:</p> <p>Unexpected error: java.security</p> <p>.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty</p> <p>java.lang.RuntimeException:</p> <p>Unexpected error: java.security</p> <p>.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty</p> <p>java.security</p> <p>.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty</p>	<p>DFC trust store is missing</p>
--	-----------------------------------

<p>[ERROR] [AGENTEXEC 2968] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,"",try_native_first, status: 0, with error message [DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "<docbroker_name>:<port>" failed</p> <p>[DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p> <p>sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p> <p>sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p> <p>[ERROR] [AGENTEXEC 2436] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,"",try_native_first, status: 0, with error message</p> <p>[DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p> <p>sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p> <p>sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target</p>	<p>The Server Certificate is missing in the DFC trust store</p>
--	---

<p>[ERROR] [AGENTEXEC 1672] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,'',try_native_first, status: 0, with error message [DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "<docbroker_name>:<port>" failed [DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>java.net.SocketException: java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.io.IOException: Invalid keystore format</p>	<p>DFC trust store is in a different format</p>
---	---

<p>[ERROR] [AGENTEXEC 2608] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,'"try_native_first, status: 0, with error message [DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "<docbroker_name>:<port>" failed [DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>java.net.SocketException: java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.security.cert.CertificateException: Unable to initialize, java.io.IOException: DerInputStream.getLength(): lengthTag=127, too big.</p> <p>java.io.IOException: DerInputStream.getLength(): lengthTag=127, too big.</p>	<p>DFC trust store is corrupt</p>
<p>[ERROR] [AGENTEXEC 5720] Detected during program initialization: Command Failed: connect,<server_name.docbase_name>,<user>,'"try_native_first, status: 0, with error message [DFC_DOCBROKER_REQUEST_FAILED] Request to Docbroker "<docbroker_name>:<port>" failed [DM_SESSION_E_RPC_ERROR]error: "Server communication failure"</p> <p>java.net.SocketException: java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.security.NoSuchAlgorithmException: Error constructing implementation (algorithm: Default, provider: SunJSSE, class: com.sun.net.ssl.internal.ssl.DefaultSSLContextImpl)</p> <p>java.io.IOException: Keystore was tampered with, or password was incorrect</p> <p>java.security.UnrecoverableKeyException: Password verification failed</p>	<p>DFC trust store contains the wrong password</p>

Recovering from a failed filestore configuration

If the configuration of a filestore to use a mapped drive as the file_system_path fails, the following error message appears:

```
DiUtil: Unable to set permission of file:  
X:\data: Unable to get DACL - Access is denied
```

Content Server is unable to connect to the drive even though you have full control on the target shared drive. This error does not occur in partitions configured as FAT32.

The reason for this failure is because of the way in which the NTFS security handles the mapped drives as compared to FAT32.

You can recover from the failure by specifying a UNC path as the file_system_path name for the filestore.

To specify a UNC path name for the filestore:

1. Log in to the Content Server repository.
2. Set the storage area offline.

```
EXECUTE set_storage_state  
WITH store = 'filestore_name', offline = TRUE
```
3. Copy the files in the storage area to the new directory.
4. Update file_system_path property of the dm_location object associated with the file store object to point to the new directory for the storage area.

```
UPDATE "dm_location" OBJECT  
SET "file_system_path" = '\\server_name\share'  
WHERE "object_name" = location_object_name
```

5. Reinitialize the server to make the change visible.
6. Set the storage area online.

```
EXECUTE set_storage_state  
WITH store = 'filestore_name', offline = FALSE
```

Note: You can also use Documentum Administrator to set the storage area offline or online.

Content Server installation directories and repository configuration scripts

This section describes the file structure, scripts, and configuration objects that are a part of a Content Server installation.

Content Server installation file structure

A Content Server installation consists of a number of files distributed among several directories. Some of these files, such as the executable files, are supplied as part of the Content Server installation package. Others, such as the Content Server startup file, are created during the installation process.

Content Server installation program or Content Server configuration program creates the subdirectories in \$DOCUMENTUM directory.

- On Windows: apptoken, cache, Checkout, config, data, dba, Export, fulltext, java64, jmsTools, local, logs, patch, presets, product, server_uninstall, share, Shared, tcf, temp, tools, uninstall, and wildfly_version
- On Linux: apptoken, cache, checkout, config, data, dba, dfc, export, fulltext, java64, jmsTools, local, logs, patch, presets, product, server_uninstall, share, tcf, temp, tools, uninstall, and wildfly_version.

uninstall

This directory contains the Content Server uninstallation program.

data

The files and directories in this category are the content storage areas. These directories must exist and location objects must be defined for them in the repository before you start Content Server. The installation procedure creates a default storage area and associated location object and a default full-text index object and associated location object.

The data directory contains the directories that store the data manipulated by users and Content Server. The installation procedure creates a subdirectory for the repository in the data directory and in that repository subdirectory, creates a content storage area.

The data includes the full-text indexes and the content files associated with objects in the repositories. The location of these directories is the most flexible component of the configuration.

Most sites will want to add more storage areas and index directories, particularly as the repository grows larger. *EMC Documentum Content Server Administration and Configuration Guide* contains information and instructions about adding additional storage areas and full-text index storage directories.

dba

The dba directory contains the log and config directories and several files.

- The log directory is where the Content Server places any log files generated by user actions during a session with the Content Server. The Content Server creates any necessary subdirectories for these log files under the log directory.
- The config directory includes a subdirectory for each repository that contains the startup files for Content Server.

fulltext

The fulltext directory contains the third-party full-text indexing software.

product

The product subdirectory contains the Content Server executables.

server_uninstall

This directory contains a script that you can run manually to destroy a repository's database tables after you delete the repository.

share

The share directory holds all the files that can be shared by the Content Server and the clients. Clients that connect to the share directory remotely can benefit in file sharing and event notification. The client must be using NFS software to receive these benefits. *EMC Documentum Content Server Administration and Configuration Guide* contains the information.

The share directory has four subdirectories:

- data

The data directory contains the data that is read and written by the Content Server and the clients. The data directory has several subdirectories. Ensure that these subdirectories can be mounted by clients.

- events

The events subdirectory contains a file for any user who has queued inbox items that have not been viewed. The files are empty. They serve as a flag to the Content Server that items that have not been viewed are in that user's inbox.

- common

The common subdirectory is where the Content Server puts copies of requested content files if users are not using client local areas and if users do not specify an alternate location for the files.

- clients

The clients subdirectory contains the win and Linux subdirectories, which respectively contain the files and executables for Windows and Linux clients.

- temp

The temp subdirectory is used by the Content Server as a temporary storage space. For example, results generated by the execution of a procedure by using the Apply method's DO_METHOD function are stored here.

- sdk

The sdk subdirectory contains two subdirectories of files that are useful to software developers. The two subdirectories are:

- Include

This subdirectory contains the dmapp.h file and the import libraries.

- example

This subdirectory contains the code examples.

Additional directories

Directory	Description
bin	Contains the Content Server software.
convert	Contains the transformation engine executable files.

Directory	Description
dba/auth	<p>When you install Content Server, a default base directory is created. Under default base directory Installer creates a subdirectory specific to the repository. The repository configuration also creates an auth_plugin location object that points to the base directory and sets the auth_plugin_location attribute in the server to the name of the location object. Any plugin installed in this directory is loaded into every server at startup for all repositories. To use a plugin only with a particular repository, place the plugin in the repository-specific dba/auth directory.</p> <p>For example, if you want to use the Netegrity plugin with a repository called engr_db, move the Netegrity module to the DOCUMENTUM\dba\auth\engr_db directory.</p> <p>When Content Server starts, it loads the plugins found in its repository-specific directory first and then those found in the base directory.</p>

Directory	Description
dba/secure/ldapdb	<p>Contains the secure connection attributes. You need to define the following setup values to use a secure LDAP connection:</p> <ul style="list-style-type: none"> • SSL mode • SSL port • Certificate database location <p>The <code>ssl_mode</code> attribute in the <code>ldap_config</code> object defines whether the LDAP server is using a secure or nonsecure connection. You need to set this when defining the LDAP setup values.</p> <p>To configure a secure connection, chose Secure as the SSL mode. When you do, the interface lets you edit the SSL port field. SSL port, represented in the <code>ldap config</code> object by the <code>ssl_port</code> attribute, identifies the port the LDAP server uses for the secure connection. This value is 636 by default.</p> <p>Certificate database location, represented in the <code>ldap_config</code> object by the <code>certdb_location</code> attribute, identifies the location of the certificate database. The attribute value is the name of the location object pointing to the certificate database. The value is <code>ldapcertdb_loc</code>. The directory that <code>ldapcertdb_loc</code> points to is <code>DOCUMENTUM\dba\secure\ldapdb</code>.</p>
example*	Contains the code examples.
external_apps	Contains a shared library.
include*	Contains the header files for any external applications that will communicate with Content Server.
install	Contains the installation scripts.
java	Contains the Java package bundled with Content Server.
messages	*.e files (error messages).
Oracle	Contains the language files needed by Oracle. During installation, the environment variable <code>ORA_NLS33</code> is set to that location. Do not remove that directory or reset the that variable.
tcf	References the task chaining framework, which is related to lifecycles and is part of the BPS and BPM group.

Directory	Description
Uniscape	Contains the NLS files for server code page conversions.
Linux*	Contains the libraries for a Linux client.
unsupported	Contains the executable files that are provided for your convenience but that are not supported by Content Server.
webcache	Includes webcache.ini. The documentation for Documentum Interactive Delivery Services/Interactive Delivery Services Accelerated contains the details.
thumbsrv	Installation directory for Thumbnail Server. The documentation for Documentum Media Services contains the details.
win*	Contains the executable files for a Microsoft Windows client. These include IAPI and IDQL for MS Windows and the DDE server and libraries.
wildfly	Contains the application server installation files used to create an instance for the Java method server.

An asterisk (*) indicates that the directory is optional.

Scripts run during installation or upgrade

During repository configuration, the following scripts are run, whether you are installing a new repository or upgrading an existing repository:

Script name	Location	Purpose	Other
headstart.ebs	\$DM_HOME/install/admin	Loads the initial default objects for the repository. Creates mount point objects, location objects, file store objects, and method objects.	
dm_apply_formats.ebs	\$DM_HOME/bin	Creates or updates format objects, which are required for content file operations.	

Script name	Location	Purpose	Other
dm_cas_install.ebs	\$DM_HOME/install/admin	Creates a method, location, template type, folder structure, and template object for use of the electronic signature.	
csec_plugin.ebs	\$DM_HOME/install/admin		
dm_routerconv_install.ebs	\$DM_HOME/install/admin	Loads methods that are used for converting routers to workflow template.	Run only during an upgrade.
templates.ebs	\$DM_HOME/install/admin	Creates default templates that are used by EMC Documentum clients for creating new documents in the repository.	
replicate_bootstrap.ebs	\$DM_HOME/bin	Creates objects and registered tables that are required for replication.	
desktop_client.ebs	\$DM_HOME/install/desktop_client	Creates folders required by Documentum Desktop and installs the default SmartList.	
disable_fulltext_jobs.ebs	\$DM_HOME/install/admin		Run only during an upgrade.
dfc.ebs	\$DM_HOME/install/admin	Loads default objects required by the Documentum Foundation Classes.	
Dfc_bof2.ebs	\$DM_HOME/install/admin	Creates the types for dm_module, dmc_jar, and dmc_java_library and configures a repository to use DFC version 5.3 SP1 and later.	

Script name	Location	Purpose	Other
dfc_javadbexpr.ebs	\$DM_HOME/install/admin	Creates types, relation types, acls, and repository folders for DFC evaluation of validation expression constraints in Java.	
dm_bpmmodules_install.ebs	\$DM_HOME/install/admin		
createMethodServerObjects.ebs	\$DM_HOME/install/admin		
csec_plugin_upgrade_53.ebs	\$DM_HOME/install/admin	Upgrades the plugin for using content-addressable storage areas.	Run only during an upgrade.
toolset.ebs	\$DM_HOME/install/admin	Installs repository administration tools.	
dm_bpm_install.ebs	\$DM_HOME/install/admin		
dm_wfTimer_upgrade.ebs	\$DM_HOME/install/admin	<p>Converts workflow pre- and post-timers set up in repositories prior to version 5.3 to the version 6.5 timer implementation.</p> <pre> dmbasic -f dm_wfTimer _upgrade.ebs -e Install -- repository _nameuserpassword </pre>	Run only during an upgrade.
dm_setup_java_lifecycle.ebs	\$DM_HOME/install/admin		
create_fulltext_objects.ebs	\$DM_HOME/install/admin	Creates repository objects related to full-text indexing.	
dm_ldap_install.ebs	\$DM_HOME/install/admin	Creates or upgrades the ldap config object type and upgrades any existing ldap config objects.	
dm_storageservices_install.ebs	\$DM_HOME/install/admin		

Script name	Location	Purpose	Other
dm_emailTemplate_install.ebs	\$DM_HOME/install/admin		
dm_xml_install.ebs	\$DM_HOME/install/admin	Installs object types and formats for XML files.	
dm_gwm_install.ebs	\$DM_HOME/bin	Executes scripts that install workflow-related types, methods, folders, and jobs.	
upgrade_java_methods_51.ebs	\$DM_HOME/install/tools	Upgrades existing Java methods.	
ci_schema_install.ebs	\$DM_HOME/install/tools	Installs the object types used by Documentum Content Intelligence Services.	
display_config_setup.ebs	\$DM_HOME/install/tools	Configures the repository for the Documentum Offline Client.	
offline_config_setup.ebs	\$DM_HOME/install/tools	Migrates offline configuration settings from the offline_config object to the docbase config object.	
upgrade_contentreplication_job.ebs	\$DM_HOME/install/admin		Run only during an upgrade.
dm_acs_install.ebs	\$DM_HOME/install/admin		
dd_populate.ebs	\$DM_HOME/bin	Populates the data dictionary with attribute and type information from datafiles.	

Configuration objects

Each repository contains the objects that together define your configuration. These objects include:

- Server config object
- Docbase config object
- Fulltext index objects

- Location objects
- Mount point objects
- Storage objects
- Format objects
- Method objects

As you make choices about how to configure the installation and repositories, modify these objects or add new ones. *EMC Documentum Content Server Administration and Configuration Guide* contains the information on configuration.

Object type categories for Oracle database storage

This section lists the object types by their size category. An object type's size category is used in two contexts:

- To determine where to create the object type's tables and indexes if the optional [FUNCTION_SPECIFIC_STORAGE] parameters are defined in the server.ini file
- To determine the default initial and next extent allotments for the object type's tables in the RDBMS

The categories for each context are not the same.

Type categories for tablespace specifications

By default, the tables and indexes for all object types are created in the same tablespace. However, you can set parameters in the server.ini file to define alternate tablespaces for large and small object types. When you do so, the system sorts the objects into large and small for the purposes of determining which object types to create in which tablespace.

The majority of the object types are considered small for this purpose. The following list shows the object types that are considered large. Any type not appearing on this list is considered small.

dm_acl	dm_process	dmi_dump_object_record
dm_assembly	dm_reference	dmi_linkrecord
dm_audittrail	dm_relation	dmi_load_object_record
dm_composite	dm_router	dmi_otherfile
dm_document	dm_script	dmi_queue_item
dm_folder	dm_smart_list	dmi_replica_record
dm_locator	dm_sysobject	dmr_containment
dm_note	dm_workflow	dmr_content
dm_procedure	dm_workitem	

Type categories for extent allocation

This section lists object type size categorizations for extent allocation and the default initial and next extent storage parameters for each category.

The object types are categorized as large, small, or default based on how many objects of the type will be created in the repository. For example, dm_document is categorized as large because most enterprises create large numbers of documents in a repository. Similarly, dm_docbase_config is categorized as small because a repository has only one docbase config object. Those types that do not fall into either the large or small category are categorized as default.

Object types categorized as large

The object types categorized as large are created with an initial extent size of 100 K. The next extent size is 1 M. The following object types are categorized as large for the purposes of allocating extents.

dm_acl	dm_reference	dmi_dump_object_record
dm_assembly	dm_relation	dmi_load_object_record
dm_document	dm_router	dmi_object_type
dm_folder	dm_sysobject	dmi_queue_item
dm_locator	dmi_containment	dmi_replica_record
dm_note	dmr_content	dmi_subcontent

Object types categorized as small

The object types categorized as small are created with an initial extent size of 10 K. The next extent size is 50 K. The following object types are categorized as small for the purposes of allocating extents.

dm_alias	dm_filestore	dm_relation_type
dm_blobstore	dm_foreign_key	dm_server_config
dm_distributed_store	dm_format	dm_store
dm_dump_record	dm_fulltext_index	dmi_change_record
dm_docbase_config	dm_linkedstore	dmi_expr_code
dm_docbaseid_map	dm_load_record	dmi_recovery
dm_extern_file	dm_location	dmi_session
dm_extern_free	dm_mount_point	dmi_sequence
dm_extern_store	dm_opticalstore	dmi_tdk_collect
dm_extern_url	dm_outputdevice	dmi_tdk_index
dm_federation	dm_registered	dmi_vstamp

Object types categorized as default

The object types categorized as default are created with an initial extent size of 20K. The next extent size is 100K. The following object types are categorized as default for the purposes of allocating extents.

dm_acs_config	dm_cabinet	dm_domain
dm_activity	dm_client_registration	dm_expression
dm_aggr_domain	dm_client-rights	dm_federation_log
dm_application	dm_cond_expr	dm_func_expr
dm_app_ref	dm_cond_id_expr	dm_group
dm_aspect_type	dm_component	dm_job
dm_audittrail	dm_cont_transfer_config	dm_job_request
dm_bocs_config	dm_dd_attr_info	dm_key
dm_builtin_expr	dm_dd_info	dm_ldap_config
dm_cabinet	dm_dms_config	dm_lightweight
dm_literal_expr	dm_public_key_certificate	dm_user
dm_media_profile	dm_query	dm_value_assist
dm_method	dm_qual_comp	dm_value_func
dm_nls_dd_info	dm-retainer	dm_value_list
dm_plugin	dm_script	dm_value_query
dm_policy	dm_smart_list	dm_validation_descriptor
dm_process	dm_staged	dm_workflow
dm_procedure	dm_type	dm_workitem
dmc_aspect_type	dmc_relationship_def	dmc_wfsd_element_string
dmc_calendar	dmc_routcase_condition	dmc_wfsd_parent
dmc_calendar_event	dmc_scope_config_relation	dmc_wfsdrp_boolean
dmc_completed_workitem	dmc_transition_condition	dmc_wfsdrp_date
dmc_datatable	dmc_type_info	dmc_wfsdrp_double
dmc_datatable_row	dmc_wf_package_skill	dmc_wfsdrp_integer
dmc_datatable_schema	dmc_wfsd_element	dmc_wfsdrp_parent
dmc_datatable_settings	dmc_wfsd_element_boolean	dmc_wfsdrp_string
dmc_module	dmc_wfsd_element_date	dmc_workqueue
dmc_preset_info	dmc_wfsd_element_double	dmc_workqueue_policy
dmc_preset_package	dmc_wfsd_element_integer	dmi_autittrail_attrs
dmi_dist_comp_record	dmi_package	dmi_registry
dm_expr_code	dmi_transactionlog	dmi_wf_timer
		dmi_workitem

Defining Oracle or DB2 database parameters for repository tables

To improve performance and increase the throughput of the system, you might want to control where repository information is stored. For example, you can store frequently used data on different disks than less frequently used data. Defining database parameters to store data in different tablespaces also partitions data into smaller, more manageable pieces.

When a repository is created, the system automatically creates object-type tables and indexes in the underlying database. The object-type tables and indexes are described in *EMC Documentum Content Server Fundamentals Guide*.

If you do an express installation of Content Server, by default, Content Server creates all object-type tables and indexes in the same tablespace. The size and number of the extents allotted for each table are determined by default configuration parameters. If you do a custom Content Server installation, you are prompted to configure the object-type tables and indexes, and you can create them in separate tablespaces.

You can edit the `server.ini` file to change any configuration parameters when the repository is created, before you start the Content Server.

- On Oracle, you can change two parameters:
 - The tablespace for the object-type tables and indexes
 - The size of the extents allotted for system-defined object types

You cannot change the number of extents allotted for the object types.

- Under Oracle 10, always create tablespaces as locally managed tablespaces (LMTs) using the LOCAL value. If you have dictionary managed tablespaces (DMTs) under Oracle 10, use the Oracle DBMS_SPACE_ADMIN package to convert DMTs to LMTs, for example,

```
SQL> exec dbms_space_admin.Tablespace_Migrate_TO_Local('Table_space1');
```

The *Oracle* documentation set contains the details on extent management and DMT-to-LMT conversion.

- On DB2, you can change the tablespace for the object-type tables and indexes. On Oracle, you can change two parameters:
 - The tablespace for the object-type tables and indexes
 - The size of the extents allotted for system-defined object types

You cannot change the number of extents allotted for the object types.

Defining the tablespace

The parameters in the [FUNCTION_SPECIFIC_STORAGE] and [TYPE_SPECIFIC_STORAGE] sections of the `server.ini` file define the tablespace in which to create the object-type tables and indexes.

FUNCTION_SPECIFIC_STORAGE

Set the parameters in the [FUNCTION_SPECIFIC_STORAGE] section to define the tablespace for the type tables and indexes for a particular category of object types. EMC Documentum sorts object types into the categories large and small for the purposes of defining their tablespace.

- Object types in the large category are those that are expected to have a large number of object instances. For example, dm_SysObject is in the large category.
- Object types in the small category are those that are expected to have very few object instances. For example, dm_docbase_config is in the small category. Each repository has only one Docbase config object.

The format of the [FUNCTION_SPECIFIC_STORAGE] section is:

```
[FUNCTION_SPECIFIC_STORAGE]
database_table_large=tablespace_name
database_table_small=tablespace_name
database_index_large=tablespace_name
database_index_small=tablespace_name
```

For example, to define a tablespace for the object-type tables in the large category, include the following lines in the server.ini file, substituting the name of the tablespace:

```
[FUNCTION_SPECIFIC_STORAGE]
database_table_large=tablespace_name
```

For example, to put the indexes for the large category in the tablespace named production_1, include the following lines in the server.ini file:

```
[FUNCTION_SPECIFIC_STORAGE]
database_index_large=production_1
```

You can specify the function-specific parameters singularly or in any combination.

TYPE_SPECIFIC_STORAGE

Set the parameters in the [TYPE_SPECIFIC_STORAGE] section in the server.ini file to define a tablespace for the type tables or indexes for a specific object type.

The format of the [TYPE_SPECIFIC_STORAGE] section is:

```
[TYPE_SPECIFIC_STORAGE]
database_table_typename=tablespace_name
database_index_typename=tablespace_name
```

You can specify the type-specific parameters individually. For example, to put the object-type tables for the dm_SysObject type the tablespace named sysobj_space, include the following lines in the server.ini file:

```
[TYPE_SPECIFIC_STORAGE]
database_table_dm_sysobject=sysobj_space
```

If you want to put both the tables and indexes for an object type nondefault tablespaces, define the tablespace for each. Defining a tablespace for an object type's tables does not affect where the type's indexes are stored. The system creates the indexes in the default tablespace. Defining a tablespace for a type's indexes does not affect where the type's tables are stored.

For example:


```
[TYPE_SPECIFIC_STORAGE]
database_table_dm_sysobject=sysobj_space
database_index_dm_sysobject=sysobj_idx_space
```

The object-type tables and indexes of any object type not specified in a type-specific parameter are created in the default tablespace or, if specified, in the tablespace for the type's category.

If the `server.ini` file includes both function-specific and type-specific parameters that apply to an object type, the type-specific parameters override the function-specific parameters. For example, suppose you add the following function-specific and type-specific parameters to the file:

```
[FUNCTION_SPECIFIC_STORAGE]
database_index_large=production_1
[TYPE_SPECIFIC_STORAGE]
database_table_dm_sysobject=sysobj_space
```

Both parameters apply to the `dm_SysObject` type because `dm_SysObject` is in the large category. The object-type tables for `dm_SysObject` are created in the `sysobj_space` tablespace because the type-specific parameter overrides the function-specific parameter.

Defining the Oracle extent sizes

For the purposes of extent allocation, the Documentum object types are sorted into three categories: large, small, and default. The category name describes the quantity of expected objects of the type. For example, `dm_document` is considered a large type because most enterprises generate large quantities of documents. In contrast, `dm_repository_config` is a small type because there is only one docbase config object in a repository. Those object types that typically do not have large numbers of objects or very small numbers of objects fall into the default category.

A type's category determines how much database storage is allocated to it by default. Object types categorized as:

- Large object types receive an initial extent of 100 KB and a next (second, third, fourth...) extent of 1 MB.
- Small object types receive an initial extent of 10 KB and a next extent of 50 KB.
- Default object types receive an initial extent of 20 KB and a next extent of 100 KB.

The default storage parameters set the initial and next extent sizes. There are also parameters that define the default minimum and maximum number of extents allotted to an object type table and the percentage increase of extents allotted after the second extent. The minimum number of allotted extents is 1 and the maximum number is determined by Oracle, based on the data block size. By default, object-type tables and indexes are allocated the minimum number of extents when they are created.

The percentage increase default is 10 percent. This means that extents allotted after the second extent are increased in size by 10 percent over the previously allocated extent. For example, if the second extent's size is 100 KB, then the size of the third extent is 110 KB, 10 percent greater than 100 KB. The fourth extent would be 121 KB, 10% greater than 110 KB.

You can change the initial and next extent default sizes for an individual object type or for an entire category by setting parameters in the `server.ini` file before the repository is created.

You can change the following parameters by using the Oracle ALTER TABLE command through sqlplus:

- Next extent
- Minimum extent
- Maximum extent
- Percentage increase

The Oracle documentation contains the instructions.

Changing storage parameters for individual object types on Oracle

To change the initial and next extent parameters for an object type, add a [TYPE_EXTENT_SIZE] section to the `server.ini` file. This section has the following format:

```
[TYPE_EXTENT_SIZE]
database_ini_ext_typeof=new_value[K|M]
database_next_ext_typeof=new_value[K|M]
```

- *typename* must be the internal name of a system-defined object type. It cannot be a user-defined object type.
- The *database_ini_ext_typeof* parameter defines the size of the initial extent allotted to the type.
- The *database_next_ext_typeof* parameter defines the size of the second extent allotted to the type.
- *new_value* is an integer. If you include K, the value is interpreted as Kilobytes. If you include M, the value is interpreted as Megabytes. If you include neither K nor M, the value is interpreted as bytes.

For example, to change the defaults for `dm_sysobject`, add the following lines to the `server.ini` file:

```
[TYPE_EXTENT_SIZE]
database_ini_ext_dm_sysobject=new_value[K|M]
database_next_ext_dm_sysobject=new_value[K|M]
```

You can set either parameter or both for an object type. The section can include parameter definitions for more than one object type. For example:

```
[TYPE_EXTENT_SIZE]
database_ini_ext_dm_sysobject=new_value[K|M]
database_next_ext_dm_sysobject=new_value[K|M]
database_next_ext_dm_user=new_value[K|M]
```

Changing storage parameters for categories of types on Oracle

To change the initial and next extent parameters for all object types in one category, add a [FUNCTION_EXTENT_SIZE] section to the `server.ini` file. This section has the following format:

```
[FUNCTION_EXTENT_SIZE]
database_ini_ext_large=new_value[K|M]
database_ini_ext_small=new_value[K|M]
database_ini_ext_default=new_value[K|M]
database_next_ext_large=new_value[K|M]
database_next_ext_small=new_value[K|M]
```

```
database_next_ext_default=new_value[K|M]
```

- The *database_ini_ext_large* parameter defines the size of the initial extent allotted by default to object types categorized as large.
- The *database_ini_ext_small* parameter defines the size of the initial extent allotted by default to object types categorized as small.
- The *database_ini_ext_default* parameter defines the size of the initial extent allotted by default to object types categorized as default.
- The *database_next_ext_large* parameter defines the size of the second extent allotted by default to object types categorized as large.
- The *database_next_ext_small* parameter defines the size of the second extent allotted by default to object types categorized as small.
- The *database_next_ext_default* parameter defines the size of the second extent allotted by default to object types categorized as default.
- *new_value* is an integer. If you include K, the value is interpreted as Kilobytes. If you include M, the value is interpreted as Megabytes. If you include neither K nor M, the value is interpreted as bytes.

For example, to change the default extent sizes for all large object types, add the following to the `server.ini` file:

```
[FUNCTION_EXTENT_SIZE]
database_ini_ext_large=new_value[K|M]
database_next_ext_large=new_value[K|M]
```

You can set any combination of the parameters. It is not necessary to set the parameters for all three categories. You can also set only one of the parameters for a category. To illustrate, the following example sets the initial extent for objects categorized as large and the next extent for object types categorized as default:

```
[FUNCTION_EXTENT_SIZE]
database_ini_ext_large=200K
database_next_ext_default=120K
```

User-defined object types

A user-defined object type derives its database storage parameters from its supertype. If the type has no supertype, then the type is assigned to the large category for tablespace assignment and to the default category for the extent allocations.

You cannot change the storage parameters for user-defined object types.

On DB2, if you create a tablespace for objects of type `dm_SysObject`, then create a user-defined object type whose supertype is `dm_SysObject`, the user-defined object type is not stored in the tablespace

for dm_SysObject. Instead, the user-defined object type is stored in the default tablespace, unless you define the tablespace for dm_SysObject in the server.ini file.

Distributed Content

This chapter describes the Content Server features that support the Documentum distributed configurations. It contains the information to help you determine which features and configurations best meet the needs of your business. Also, it provides procedures to implement and manage those features and configurations.

The information is intended for system administrators or superusers who are responsible for the implementation and maintenance of a Documentum distributed environment.

Distributed Configuration components

This section describes the Content Server features that are the building blocks for the common distributed configurations.

Building blocks

This section describes the features that are the building blocks for implementing distributed configuration models.

Network locations

Network locations are a basic building block of a single-repository distributed environment for web-based clients. Network locations represent a place or area on network's topography. A network location can represent:

- A branch office
- A group working in the same geographical area
- Any set of users you choose to aggregate as a network location

Network locations typically identify one or more IP addresses or address ranges. The addresses generally represent machines that are close to each other. For example, all machines in one branch office could be defined as one network location. Another network location could represent all users in the Western United States or in Eastern Europe. The administrator who configures the system determines the geographic size of a network location.

Benefits and best use

Network locations are useful only when configuring a single-repository distributed environment for web-based clients. Desktop-based clients do not use network locations.

Users connecting to a repository through a web-based client are automatically connected to the closest server for content requests by using network locations. Network locations enhance performance when the users access documents and any other object with content.

Configuration requirements

To use network locations, fulfill the following configuration requirements:

- Designate one repository as the global registry, and you must store the network location definitions in that repository.
- Specify a Content Server or ACS server's proximity value for each network location.

A Content Server or ACS server's proximity value defines the network location's proximity to those servers. This information is used to ensure that the server closest to the user manages content requests.

ACS servers

The An ACS (Accelerated Content Services) server is a lightweight server that handles read and write content operations for web-based client applications. There is one ACS server for each Content Server host installation. It communicates with one Content Server for each repository in the host installation. The ACS server uses HTTP or HTTPS for all communications.

The ACS server is installed automatically when the first repository on the installation is configured. If you add repositories to the installation, the ACS server configuration information is updated so that the server can communicate with the primary Content Server for the new repository.

If you install a remote site with a remote Content Server, then the installation at that site also has its own ACS server.

Note: ACS does not use nor require user authentication. Furthermore, if user authentication (including Kerberos and other supported single sign-on standards) is implemented in the Documentum system, then as long as a user successfully authenticates through their Web client, the appropriate ACS server seamlessly returns the requested content.

Benefits and best use

The ACS server serves users who are accessing the content through web-based client applications.

Note: You cannot use an ACS server to handle content read and write requests for users on desktop client applications.

Configuration requirements

Configure ACS servers as follows:

- At least one valid ACS configuration object for the ACS server in each repository served by that ACS server must exist.
- Configure the `acs.properties` file for the ACS correctly. (The installation process automatically configures an `initialacs.properties` file.)
- To enable asynchronous write operations for the server, set the `acs_write_mode` property in the content transfer configuration object in the repositories correctly.
- Ensure that the ACS server ports are open if the network locations that the server is servicing are outside a firewall.
- If the Content Server and application server clocks are out of sync by more than six hours (the default), URLs expire. When URLs expire, content transfer for uploading and downloading of content fails. Therefore, synchronize the Content Server and application server clocks with a time server. You set the time interval after which URLs expire in the `acs.properties` file on the ACS host.

ACS caching

You can configure the ACS server to perform content caching. To set up content caching, modify properties as mentioned.

- Windows:

```
%DOCUMENTUM%\wildfly_directory\server\DctmServer_MethodServer\
deployments\acs.ear\lib\configs.jar\config\acs.properties
```

- UNIX:

```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_MethodServer/
deployments/acs.ear/lib/configs.jar/config/acs.properties
```

To set up ACS caching:

1. Set the `acs.cache.enabled` property to `true`. This property enables content caching. By default content caching is disabled.
2. Set the `cache.store.root` property to the location where you want to store the cached content, for example, `C:\...\Documentum\acsCache`. This property sets the root directory for the cache that holds the content.
3. Set the `cache.store.quota` property to `1000M`. This property determines the quota size of the cache.

BOCS servers

A Branch Office Caching Services (BOCS) server is a caching server. It is a separate, optional product with its own installer. It is not installed with Content Server.

BOCS servers cache content locally. When a BOCS server handles content requests from users, it caches the requested content locally. BOCS servers can pre-cache content through a pre-caching job or programmatically. Caching content allows users to obtain frequently accessed content quickly. You can configure the amount of content that is cached and the length of time for holding the content.

When users save content to the repository through a BOCS server, the underlying client application determines whether the write operation is asynchronous or synchronous. This application might default to one or the other or offer users a choice between synchronous or asynchronous write operation. If the content is written asynchronously, it is cached on a BOCS server until an internal job runs to save the content to the repository.

BOCS servers communicate with ACS and DMS servers instead of directly with Content Servers. You can configure the repositories that a BOCS server serves.

Additionally, you can configure a BOCS server in either push or pull mode to obtain messages from a DMS server. In push mode, a BOCS server accepts messages sent to it from a DMS server. In pull mode, a BOCS server contacts the DMS server and takes messages from the DMS server's message queue. If there is a firewall between the DMS server and the BOCS server, the BOCS server is typically configured in the pull mode. If there is no firewall between the two servers, the BOCS server is typically configured in push mode.

BOCS does not use nor require user authentication. Furthermore, if user authentication (including Kerberos and other supported, single sign-on standards) is implemented in the Documentum system, then as long as a user successfully authenticates through their Web client, the appropriate BOCS server seamlessly returns the requested content.

Note: If your SSO implementation requires cookie authentication in order to connect to the BOCS host, then all hosts (Web client, UCF client/application server, BOCS, and Content Server/ACS) must use the same SSO server because the UCF client uses the Web client's authentication cookie, which is provided by an SSO server, when connecting to the BOCS host.

Benefits and best use

Use BOCS servers to provide web-based clients the fastest possible access to frequently used content without having to install a distributed storage area remote site.

Limitations

Access to the file systems on the BOCS server host can be less secure than access to content storage areas in the repository.

BOCS encryption

BOCS servers support content encryption. Content encryption enhances the security of confidential content by preventing the unauthorized access and viewing of content. You can select from the following encryption options:

- You can allow the BOCS server to decide whether to encrypt content.
- You can configure the BOCS server to encrypt content always (also known as “unconditional encryption”)
- You can disable content encryption for the BOCS server.

Partial download of content

BOCS servers support partial download of content. BOCS servers can serve content as it arrives without having to wait for the remainder of the content file to arrive. This practice improves performance because users can start viewing content sooner than waiting for the entire content file to download.

The Documentum Administrator online help contains more information.

Configuration requirements

Configure a BOCS server as follows:

- A BOCS server configuration object representing the BOCS server must reside in the global registry.
- Configure an `acs.properties` file correctly for each BOCS server. The installation process configures an `acs.properties` file automatically.
- The ports on which BOCS server serves content to users must be open.
- If the BOCS server is configured in pull mode, the server must have access to the URL defined in the `message_consume_url` property of the DMS configuration object.
- If the BOCS server is configured in push mode, the DMS server must have HTTP access to the BOCS server.
- To use the pre-caching feature, enable pre-caching in the content transfer configuration object in each repository for which you want the feature enabled.

Note: The pre-caching feature is enabled by default.

DMS servers

A DMS server is a server that routes messages between BOCS and Content Server instances. A DMS server routes the following messages:

- Content pre-caching
- Asynchronous write operations

You can deploy DMS servers in a high-availability configuration.

Benefits and best use

The operations of a DMS server are integrated into the distributed environment. The installation is simple and minimal configuration is required.

Pre-cached content

Pre-cached content is content that has been cached on a BOCS server before users request that content. You can cache the most up-to-date version of content that users frequently request before they request it again.

Benefits and best use

Pre-caching content is most useful for large content files that are frequently or regularly requested by users. It provides the fastest possible content delivery to users because the content is already cached as close as possible to the users.

Limitations

- You cannot not pre-cache virtual documents or content that requires manipulation before viewing.
- Additionally, resource forks of Macintosh files are not pre-cached; only data forks are pre-cached.

Configuration requirements

Using content pre-caching has the following configuration requirements:

- Ensure that the DMS server is correctly configured.
- Ensure that content pre-caching is enabled in the content transfer configuration object in the repository that contains content you want to pre-cache.

Content pre-caching is enabled through Documentum Administrator. The Documentum Administrator online help contains more information.

- Specify the content to pre-cache, either through a pre-caching job or programmatically through DFC.

Asynchronous write capabilities

Asynchronous write means to store content on a BOCS server only and then write that content to the repository later. Metadata is still written immediately to the repository. Even after content is written to the repository, the content remains in the BOCS server's content cache.

Note: In contrast to asynchronous write, synchronous write means to write content immediately to the repository. The underlying application can default to one type of write or allow the user to choose one.

As long as the content resides on the BOCS server only, users who do not have access to the BOCS server cannot access the content for viewing or modification nor can they use administration methods, such as `MIGRATE_CONTENT`, to access the content.

Requests to write content to the repository are sent to the BOCS server where they are processed in first-in-first-out (FIFO) order. If the request is not executed immediately, then a user-defined job sends the request again. This job implements the `dm_AsynchronousWrite` method, which checks the relevant metadata on the Content Server to determine which content is parked on the BOCS server. For each piece of content that is found, a message requesting to write that content to the repository is sent to the DMS server.

Benefits and best use

Asynchronous write operations ensure that a user does not wait for content to be saved to the repository when the network communication is slow. Additionally, other users in the network locations served by the BOCS server on which the content is stored have immediate access to the content.

Asynchronous write operations are best used in the following situations:

- The branch office and primary office have slow network connections.
- When users at the network locations served by the BOCS servers primarily use that content.
- The content is large.

Limitations

Using asynchronous write has the following limitation:

- Content is unavailable to users who are not accessing the repository through the BOCS server on which the content is stored.

Configuration requirements

Using synchronous write has the following configuration requirements:

- In the content transfer configuration object, configure the write mode for the ACS server to allow write operations.

Configure the BOCS server for asynchronous write as follows:

- Enable the BOCS server for asynchronous write.
- Enable asynchronous write in the content transfer configuration object in the global registry.
- Ensure that the DMS server is configured appropriately.
- There must be an index on the `i_parked_state` and `r_object_id` properties of the `dmr_content` object in the database.
- Set the `dm_AynchronousWrite` job to the active state.

This job is installed in the inactive state.

Distributed storage areas and configuration requirements

A distributed storage area is a single storage area made up of multiple component storage areas. They are the foundation of the single-repository distributed model for desktop clients. You can also use a distributed storage area in web-based models. All sites in a model using a distributed storage area share the same repository. However, each site has its own local storage area component to provide fast, local access to content.

The component storage areas can be file store or linked store storage areas. If you encrypt one component, encrypt all components. It is not possible to have a distributed storage area with some file-store components that are encrypted and some that are not encrypted.

Note:

- Linked store storage areas are not supported.
- A file store storage area may not use compression and de-duplication if that storage area will be a component of a distributed storage area. Distributed storage areas do not support compression and de-duplication.

You can either share content files or replicate them between component storage areas.

Benefits and best use

Using a distributed storage area solves the performance problems experienced by remote desktop users when all sites share one repository with centralized content storage. For example, if the repository and its content files are located in Nice, France, then users in Toronto, Canada, can experience delays in accessing files due to the distance between them and the repository. If you set up a distributed storage area, each site has fast access to content files in a local storage area. Users in Toronto no longer experience poor performance when opening documents.

For web-based users, distributed storage provides an alternate configuration model if the preferred model is not acceptable.

Limitations

After a repository begins using a distributed storage area, it is not possible to remove that storage area and return to a standalone, non-distributed configuration.

Configuration requirements

Configure a distributed storage area as follows:

- The host machines for the participating servers must run the same operating system.
- Your network must have high bandwidth and good reliability to support repository access.
- All the components of the distributed storage area and the containing distributed store object must have the same value in the `media_type` property. This property indicates whether the files stored in the storage area are thumbnail renditions, streaming content, or another kind of content.
- Install the index agent and index server only at the primary site.

Additional requirements depend on how you choose to handle content files.

Remote Content Servers

A remote Content Server (RCS) resides at each remote site that has a component of a distributed storage area. Remote Content Servers are automatically configured to provide maximum performance for desktop users for content-related queries. Remote Content Servers do not handle metadata requests.

When a repository is configured with a distributed storage area, there is a primary site, and one or more remote sites. The RDBMS, which holds the repository's metadata, resides at the primary site. The Content Server at the primary site is configured to service all metadata requests, as well as content requests from clients local to that server. The remote Content Servers at the remote sites are configured to handle only content requests. These servers provide content to users from their local storage area and write content to that storage area. An RCS does not handle metadata requests.

This model provides reasonably good performance for content requests because content is stored locally and accessed by a local server. It also provides good performance for metadata requests because the server closest to the RDBMS manages the requests.

Benefits and best use

Remote content servers provide increased performance for repository queries when a repository has a distributed storage area that is distributed across different geographical sites.

Limitations

- You cannot use an RCS as a failover server for the primary server.
- The Content Server at the primary site cannot fail over to an RCS.
- The primary Content Server and remote Content Servers must be of the same version.

Configuration requirements

Configure a distributed storage area and its remote content servers as follows:

- Proximity values for the Content Server at each site must identify one server as the metadata server and all others as remote Content Servers.

You can still modify the basic proximity values even though they are configured automatically when the sites are installed.

- Install the index agent and index server at the primary site.
- Set the metadata server's session timeout value to a minimum of 30 minutes.
- Configure servers at all sites to use the same authentication mechanism to authenticate all remote users and groups accessing distributed documents.
- The remote Content Servers and the data server must have compatible secure connection mode configurations. This configuration ensures that clients connecting from remote sites can access the remote content server and the metadata server. Alternatively, the secure connection default for the client must allow the client to request a connection on a native or secure port.

Configuring the servers to accept either secure or native connections is the easiest solution.

- It is recommended to provide unique hostnames in remote Content Server configuration when more than one host is used.

Shared content

Shared content files are files that are stored only in one component of a distributed storage area but are still accessible to users at all sites. The remote Content Servers fetch shared content files directly

when needed. Content Servers can fetch the content files directly when a distributed-storage area's components are configured as shared drives.

Benefits and best use

Using shared content eliminates the need to enable surrogate get functionality or run `ContentReplication` jobs at each component site.

In a distributed storage area, the best documents for sharing are the ones local to one site that users from other sites do not access frequently.

Configuration requirements

Configure shared content files as follows:

- Configure each distributed-storage area's component to be a shared directory.
- Configure the installation owner (`dmadmin` or equivalent) at each site to be the same account at all sites.
- On Windows platforms:
 - All the host machines must belong to the same domain.
 - The Content Server installation owner must be a domain user and administrator in that domain.
 - Specify the UNC path for the location objects representing the shared storage areas.
 - The Content Server installation and the repository owners must have Full Control permissions for the storage locations.

Content replication

Content replication supports configurations with distributed storage areas. In a distributed storage area within a repository, a portion of the content files stored in each site are replicated to other sites. Because each site has a copy of the content files, servers accessing these files do not fetch them from a remote storage area.

Content Server provides automatic replication, through the `ContentReplication` tool, on-demand replication, using the surrogate get feature, or manual replication, using the `REPLICATE` or `IMPORT_REPLICA` administration methods.

Note: Content replication does not replicate content from one repository to another one.

Benefits and best use

Replicated content optimizes content transfer performance when users open a document because access is local. Consequently, the best candidates for replication are documents that are large or accessed frequently by users at all locations.

Configuration requirements

The `ContentReplication` tool and the surrogate get feature both require a homogenous server host environment. Therefore, the host machines for all of the participating servers must be all Windows or all UNIX machines.

If you use the `REPLICATE` administration method to replicate content or the `IMPORT_REPLICA` method to copy content, configure the servers to connect with each other. On UNIX platforms, the servers must be able to connect using NFS.

The secure connection mode setting of the target server must be compatible with the connection mode requested by the client performing the content replication.

Reference links

Reference links are a feature of a multi-repository configuration. If your deployment has multiple repositories, users are not limited to working only with objects in the repository into which they logged in. Users can also work with objects in other repositories. For example, a user might receive a task in their home repository inbox with an attached document that resides in a different repository. Or a user might view or update an object in a repository that is a replica of an object in a different repository.

A reference link in one repository points to an object in another repository. A Content Server creates reference links as needed when users work with objects in multiple repositories or when object replication occurs. The following operations create reference links:

- Linking a remote object to a local folder or cabinet
- Checking out a remote object
- Adding a remote object to a local virtual document
- Object replication jobs

Benefits and best use

Reference links simplify work sessions for users. Users can start one repository session and manipulate objects in any repository without starting a new session each time they want to access an object in another repository. Reference links can also provide users with local access to the properties of a remote object.

Configuration requirements

To enable the reference links feature:

- Users must have accounts in each repository they access.
- Each participating repository must project to the connection brokers of the other participating repositories.
- If you installed any Content Servers in the participating repositories with trusted server licenses, configure all servers to listen on a secure and a native (unsecured) port. The Documentum Administrator online help contains the instructions.

To facilitate user access, you can configure both or one of the following:

- Add all of the repositories into the same federation
- Use an LDAP directory server to manage users and groups

A federation is a Documentum feature that facilitates management of global users and groups and external ACLs across repositories. Any addition or deletion of a global user or group or modification of a global user or group property is made in the governing repository. The governing repository then propagates the changes to the member repositories. External ACLs are automatically replicated to member repositories as needed so that security for the global users and groups is uniform within the federation.

An LDAP directory server is a third-party product that provides a single place for maintenance of some or all users and groups in your enterprise. User and group entries are created in the directory server and those entries are propagated to all repositories that are set up to use the directory server. The property information that is propagated is defined when you set up the repository to use the LDAP directory server. This property information is not limited to the global properties of the users and groups.

Unlike a federation, the LDAP directory server does not replicate external ACLs to participating repositories. If you use a directory server without a federation, then manage ACL replication manually.

If you use a federation and an LDAP directory server, then the directory server communicates with the governing repository in the federation. It also communicates with any other nonfederated repositories with which you want to use the directory server. The governing repository propagates the user and group changes it receives from the LDAP directory server to the member repositories. It also manages the external ACLs within the federation.

Users who are managed by an LDAP directory server do not require an operating system account. However, Content Server requires an operating system login name to create the user. This login must be unique within the system. For LDAP-managed users, you can define operating system names for them without actually creating the operating system accounts. The Documentum Administrator online help contains the information on configuring a repository to use an LDAP directory server.

Object replication

Object replication supports multi-repository, distributed configurations. These configurations have multiple sites with a separate repository and relational database (RDBMS) at each site. Object replication replicates entire objects (property data and content) between repositories.

In the target repository, replicated objects are stored as replica objects with associated `dm_reference` objects. Each replica object has an associated reference object. With only a few constraints, users can manipulate replica objects much as they do source objects. Users can review, annotate, or index replicas. They can query against them or add them to a virtual document. They can also manipulate the source object through the replica.

Jobs automate replication and they are defined by the business requirements of the enterprise.

Benefits and best use

Replication can reduce network traffic because users access local replicas of the objects for most operations. Replication is of most benefit during peak usage times.

Use object replication if you want local autonomy; for example, if your wide area network (WAN) is not always reliable. If objects are replicated into a target repository, then users can continue to work even if the replica object's source repository is not available.

Configuration requirements

You enable object replication as follows:

- All participating sites must project to the connection brokers at all other participating sites.

Note: Cross-projection between all sites is only required if you want:

- To allow users to manipulate source objects through the replicas
- To allow the server at the target repositories to perform automatic replica refreshes

If your replicas are read-only, then cross-projection is not necessary. For example, if you are replicating from a source repository inside a firewall to a target repository outside a firewall, then the target server does not need to project to the source repository's connection broker nor the source to the target.

- Because object replication uses the dump and load operations, the databases of the source and target repositories must either use the same code page, or the target database must use unicode. For example, if the source database is using Japanese and the target is using unicode, the replication operation succeeds. However, if the source is using Unicode and the target is Japanese, replication fails.
- The `secure_connect_mode` defined for the server in the target repository must be compatible with the secure connection mode requested by the content replication job that performs the object replication.

The server's secure connect mode is defined in the server configuration object. The mode that the job requests is defined in the `dfc.session.secure_connect_default` key in the `dfc.properties` file used by the job. That file is found on the host machine where the job resides. The mode defaults to `try_native_first`.

The Documentum Administrator online help contains the instructions on setting `secure_connect_mode` for a server or a client application.

- If you are replicating documents created on Macintosh client machines, all participating sites must use the same Macintosh access protocol.
- Both the source and the target sites for each replication job must have enough disk space to accommodate the temporary dump files created by the job.
- Each target site must have enough disk space to accommodate the replicated content files.
- Network latency affects replication performance. For adequate performance, a ping between any two participating sites should be 200 milliseconds or less.

Federations

A federation is a set of two or more repositories that are bound together to facilitate the management of a multi-repository distributed configuration. One repository in the set is the governing repository. The remaining repositories are member repositories.

One enterprise can have multiple federations, but each repository can belong to only one federation.

A federation can include repositories with both trusted and non-trusted Content Servers.

Benefits and best use

In a multi-repository distributed configuration, users are typically working with objects from more than one repository in the same session. The objects can be the original objects, mirror objects, or replica objects. To maintain data consistency, object type and format definitions must be the same across all of the repositories. Similarly, to maintain consistent security, the definitions of users, groups, and ACLs must be the same across all of the repositories.

If data and security consistency is maintained manually in each repository, then keeping objects synchronized in multiple repositories can be time consuming and error prone. A repository federation automates much of the management of data and security consistency. The governing repository automatically propagates changes you make to users, groups, and external ACLs to each repository in the federation.

Federations are best used in multi-repository production environments where users share objects among the repositories. It is not recommended to create a federation that includes a mixed environment of production, development, and test repositories because development and test repositories can be different versions than the production ones, contain object type and format definitions that change frequently, and have a small subset of all of your users.

Configuration requirements

For the most consistent behavior in a repository federation or any multi-repository, distributed configuration, follow these configuration requirements:

- Object type and format definitions must be the same across all participating repositories.

Because Documentum Administrator does not propagate type or format changes in a governing repository to member repositories, perform this operation manually or by using your own applications.

- Users and groups must be the same across all participating repositories.

Documentum Administrator is used to manage all global users and groups in a repository federation. At setup, define whether users and groups are global or local. Making them all global is recommended.

The federation update jobs automatically propagate all changes to global users defined by `dm_user` objects. If you define users by subtypes of `dm_user`, they are not automatically propagated unless you identify those subtypes in the federation's properties. You can do that when you create the federation or after, as a federation modification.

Changes to global users and groups are only allowed using the Documentum Administrator and are propagated automatically to all member repositories.

- The Content Server at the governing site must project to the connection brokers at the member sites.
- The Content Servers at the member sites must project to the connection broker at the governing site.
- If you installed any of the participating Content Servers with trusted server licenses, ensure that either:
 - The servers are configured to listen on both a secure and a native port
 - The secure connection default for clients allows the clients to request a connection on a native or secure port

Configuring the servers to accept either secure or native connections is the easiest solution. The Documentum Administrator online help contains the instructions on setting the Content Server's connection mode.

Note: When you set up a federation and use an LDAP server for accessing users, the governing repository synchronizes users and populates the `user_login_domain` with the name of the LDAP configuration object. For example if an LDAP configuration object is called `MS_LDAP`, this value is populated on the user domain. When users are then synchronized to members as part of the federation update, those same user attributes are populated. However, if on the member, the LDAP configuration name is different, authentication fails. Therefore, if you are using an LDAP configuration object to synchronize users on the governing repository, each member must have an LDAP configuration object with the same name.

Distributed Configuration models

This section provides information on configuration settings that affect Documentum CMIS, including JVM, Linux, and application properties settings.

Overview of models

You can set up a distributed configuration for either a single repository or multiple repositories. In a single-repository distributed configuration, content is distributed across all sites configured to access one repository. In a multi-repository distributed configuration, objects (content plus metadata) are distributed across all participating repositories.

The most common distributed configurations are as follows:

- For single-repository distributed environments:
 - A single repository with content stored at the primary site and accessed from remote sites using an Accelerated Content Services (ACS) server. Optionally you can also use a Branch Office Caching Services (BOCS) servers
 - A single repository with content stored in a distributed storage area and accessed from remote sites using remote Content Servers, ACS servers, and optionally, BOCS servers

[Single-repository distributed models, page 221](#) contains more information.

- For multi-repository distributed configurations:
 - Multiple repositories that replicate objects among themselves
 - Multiple repositories organized as a federation

[Multi-repository distributed models, page 249](#) contains more information.

An enterprise can use one model or a combination of the models. For example, you might have one repository that contains primarily material, such as SOPs, that users only read. You might configure that repository to use BOCS servers for its remote users. Another repository might store business and sales proposals that are written and updated frequently. That repository uses distributed storage areas with ACS servers or desktop clients for remote users. You might also tie together the two repositories in a federation, to ensure that the users and groups in both repositories are the same.

Single-repository distributed models

In a distributed single-repository, the distributed data is content. Users in many locations want fast access to the documents, that is, content, in the repository. EMC Documentum supports distributed content models for both web-based and desktop access.

Note: The figures in this section depict geographic locations, not individual machines.

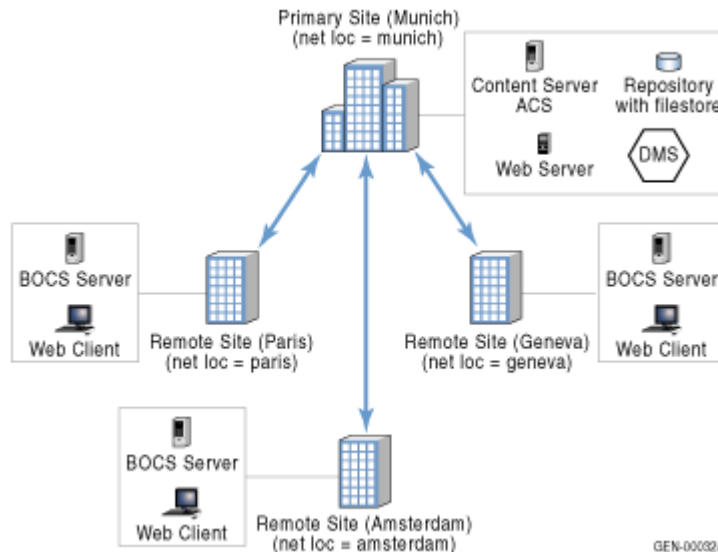
Single model 1: Single repository with content persistently stored at primary site and accessed using ACS or BOCS servers

In this model, remote users connect through a web browser. The content is stored at the primary site and content operations are handled through either an ACS or BOCS server. The ACS server is a Content Server dedicated to handling content. It does not process metadata. It only handles read and write content requests. The BOCS server is a separate product. It is a caching server that communicates only with ACS servers. Like the ACS server, it does not handle metadata requests. Both the ACS and BOCS servers use HTTP or HTTPS protocol to process client content requests.

Single model 1 is the preferred model when remote users are accessing repository content through a web-based application.

Two alternative configurations for this model exist. The configurations differ in how users at remote sites access the content at the primary site. In the first alternative, remote sites have a BOCS server installed and clients at each remote site use that BOCS server to access content. In the second alternative configuration, users at remote sites have only web clients, and they access content using the ACS server at the primary site.

Figure 10. Alternative 1: BOCS Servers at Remote Sites Communicating with Primary Site

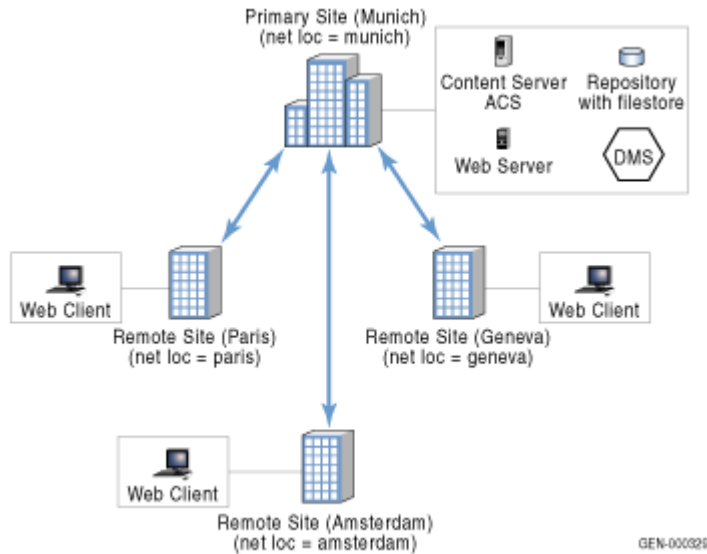


In this example of alternative 1, users at each remote site use a BOCS server to handle content requests. The BOCS server in the Paris branch office manages requests from Paris users. The BOCS server in the Amsterdam branch office manages content operations for users in the Amsterdam branch office. The BOCS server in the Geneva branch office manages requests from Geneva users.

The BOCS server is a caching server. It maintains a cache of content files requested by users. You can also pre-cache content on a BOCS server. For example, you can cache content that users access frequently or regularly on the server before users request that content. You can perform pre-caching by a job or programmatically.

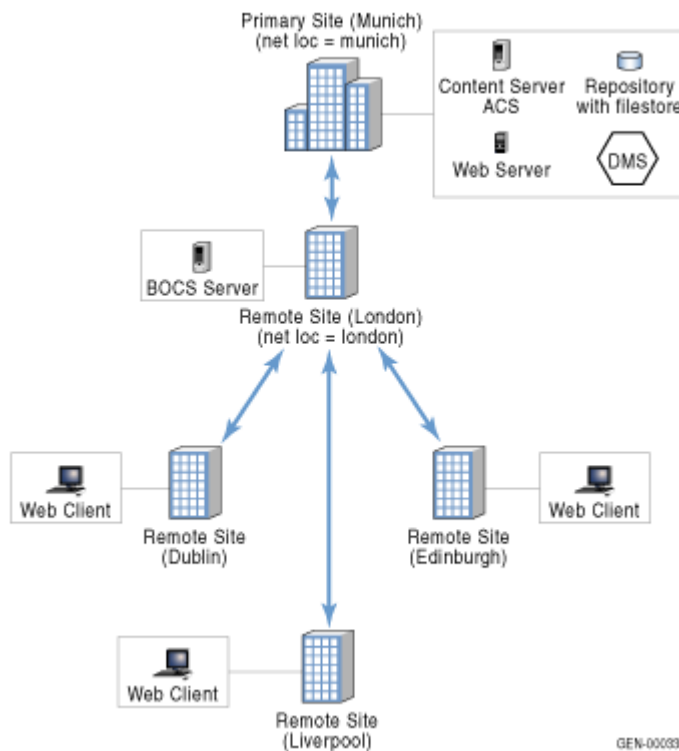
When the BOCS server receives a request for content, it first checks the cache that it maintains. If the content is in the cache, the BOCS server provides that content to the user. If the content is not in the cache, the BOCS server communicates with the ACS server at the primary site to locate the content. The ACS server, in turn, communicates with the web server and Content Server at the primary site.

Figure 11. Alternative 2: Remote sites, without BOCS servers, using primary site's ACS server



In the second alternative, the ACS server at the primary site manages requests content operations from users at a remote site.

The illustration describes variation of the configurations that combines the two. In this example, the remote clients are telecommuters, working from their web browsers on home machines. When they request documents, a BOCS server installed at the branch office closest to their location manages the request. Content Server at the primary site manages all metadata requests.

Figure 12. The two alternatives for single model 1 combined

In each of the configuration alternatives, each ACS or BOCS server is defined as a network location. When users begin a web-based session, the client application determines which network locations are available to the user. Users are either automatically assigned to a network location or they can choose from available locations, depending on how the client applications are configured. The network location with which the session is associated determines which ACS or BOCS server handles each user's content requests.

Benefits and best use

This model requires the least amount of administrative overhead. It is easy to set up and configure, and ongoing administration needs are minimal. There are no content replication jobs to define, administer, and run.

This model is not available for users on desktop clients.

Single model 2: Single repository with content in a distributed storage area

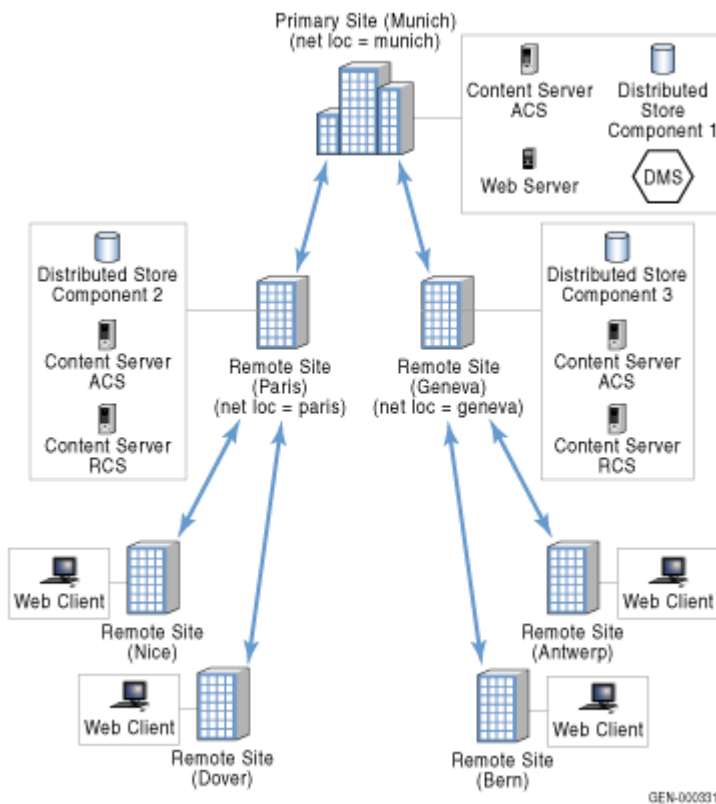
In this model, content is stored in a distributed storage area. A distributed storage area is a storage area with multiple component storage areas. One component is located at the repository's primary site. Each remote site has one of the remaining components. Each site has a full Content Server

installation: a remote content server (RCS) and an ACS server for the repository. Content is replicated from its source component to the remaining components by user-defined content replication jobs.

You can use this model for either web-based clients or desktop clients.

If a remote site is using web-based clients, the site must configure the use of an ACS server to manage content. In addition to configuring an ACS server, you can also configure a BOCS server. Also install a DMS server to facilitate pre-caching for BOCS servers and asynchronous write operations for remote users. Desktop clients at remote sites use the Content Server at the remote site to access content. In this configuration, Content Server at the primary site manages metadata requests, and an RCS or ACS at the remote sites manage content operations.

Figure 13. Single Model 2: Single repository with a distributed storage area



In this model, users in the branch offices in Nice and Dover access content stored in distributed storage at the larger Paris branch office. Users in the branch offices at Bern or Antwerp access content stored in the Geneva branch office. If users are logging in using a web-based client, ACS manages content requests at the appropriate branch office in Paris or Geneva. If users are logging in using a desktop-based client, Content Server in Paris or Geneva manages content requests. In this model, the remote sites using web-based clients could use BOCS servers for managing content requests, instead of web browsers.

Benefits and best use

For sites using web-based clients, this model is best if model 1 is not acceptable for any particular reason. For sites using desktop clients, this model is the only model available for a single-repository, distributed configuration.

Building block architectures for single-repository models

The distributed models for web-based clients use either an ACS server or a BOCS server (or both) to distribute content to users at remote sites. These models use the network location, ACS server, BOCS server, and proximity values building blocks.

The distributed model for desktop-based clients uses the distributed storage area, shared or replicated content, and proximity values building blocks.

This section describes how the various building blocks work to support the distributed model for a single repository. It also includes a description of the internal communications in the web-based model.

Network locations

Network locations are recorded in the repository designated as the global registry. They are stored as `dm_network_location_map` objects. The properties in the object record the following information:

- The name of the network location
- The IP addresses or address ranges are assigned to that location

Use network locations to determine an end user's proximity to a Content Server or ACS server. Define network locations for remote sites with BOCS servers or ACS servers. The definitions are not required if remote sites are using the ACS at the primary site and there are no servers installed at the remote site.

ACS servers

ACS servers handle content requests from web-based clients. There is one ACS server for each Documentum installation on a server host machine. The ACS server runs in the Java Method Server.

When you configure the first repository in an installation, the procedure installs a Content Server for the repository and an ACS server for the installation. If it does not exist and deploys the `acs.ear` file, the procedure also configures the Java Method Server service.

If you later configure additional repositories in that installation, the procedure automatically updates the ACS configuration information for the existing ACS server. Adding additional repositories does not add additional ACS servers.

Two internal components enable you to configure an ACS server: the `acs.properties` file and ACS configuration objects. Each ACS server has one `acs.properties` file and at least one ACS configuration object. The file is created when the ACS server is installed. The ACS configuration object is created when a repository is configured.

The `acs.properties` file identifies the repositories with which the ACS server communicates and is located in the following directories:

- Windows:

```
%DOCUMENTUM%\wildfly_directory\server\DctmServer_MethodServer\
deployments\acs.ear\lib\configs.jar\config
```

- UNIX:

```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_MethodServer/
deployments/acs.ear/lib/configs.jar/config
```

The ACS configuration object resides in a repository. Each repository served by an ACS server must have an ACS configuration object for that ACS server.

The properties of the ACS configuration object configure various behaviors for the server. They can record the proximity values that define the ACS server's distance from each of the locations. The ACS configuration object also has a property that identifies with which Content Server the ACS server communicates in the repository.

Note: If there are multiple servers installed on a single host machine for a single repository, the ACS server communicates only with one of the servers. You cannot manually configure ACS configuration objects for additional servers on a host machine.

The ACS configuration object can also identify which storage areas the server can access. These storage areas must be either standalone file store storage areas or file stores that are components of a distributed storage area. An ACS server cannot access content in other types of storage areas. Nor can it directly access encrypted or compressed content or virtual documents or XML documents. If an ACS server receives a request for a document that it cannot access, the request is forwarded to its associated Content Server. That Content Server services the request and sends the result back to the ACS.

BOCS server

The BOCS server configuration is recorded in an `acs.properties` file and a BOCS configuration object.

The `acs.properties` file for a BOCS server configures the location of the server's content cache. It also defines how long the server holds the content and other behavioral characteristics of the server. In a BOCS installation on a client host, the file is placed in a location dependent on the host operating system as follows:

- Windows:

```
%DOCUMENTUM%\wildfly_directory\server\DctmServer_BOCS/
deployments/bocs.ear/lib/configs.jar/config
```

- UNIX:

```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_BOCS/  
deployments/bocs.ear/lib/configs.jar/config
```

The BOCS configuration object identifies the network locations that the BOCS server serves and allows you to configure the following characteristics:

- Whether to run the BOCS server in push or pull mode
- The repositories that the BOCS server can service
- Whether the BOCS server can perform asynchronous write operations

Store the BOCS configuration object in the global registry repository and modify it by using Documentum Administrator.

Push and pull modes

A BOCS server can communicate with a DMS server in either push or pull mode. This configuration is set in the BOCS configuration object. If the BOCS server is configured in push mode, the DMS server sends messages routed to the server through DMS to the BOCS server. In such cases, the DMS server must have HTTP access to the BOCS server.

If the BOCS server is configured in pull mode, the BOCS server picks up messages routed to the server through DMS. The DMS server does not send them to the BOCS server. In such cases, configure the URL that the BOCS server uses to contact the DMS server in the `acs.properties` file associated with the BOCS server. Additionally, the BOCS server must identify itself to the DMS server by using the PKI credentials installed with the BOCS server. You can configure PKI credentials in the BOCS configuration object.

You can change the push or pull setting for a BOCS server. However, if you change from pull mode to push mode, the messages pushed to the BOCS server might not arrive if a firewall exists between the DMS server and the BOCS server. Additionally, queued messages in the DMS server are marked as either push or pull messages. Consequently, if you change from pull mode to push mode, the BOCS server does not receive any queued messages that are marked as pull messages. Pull mode, rather than push mode, is typically used for BOCS servers outside of a firewall.

Repository inclusion or exclusion

You can configure which repositories a BOCS server can service by defining either a repository inclusion list or a repository exclusion list. Two underlying properties record this configuration: `is_inclusion_list` and `docbase_names`. If you choose to specify the repositories that the BOCS server can service, then the `is_inclusion_list` property is set to `T`. The repositories listed in `docbase_names` are the repositories that the BOCS server can service. If you choose to specify the repositories that the BOCS server cannot service, `is_inclusion_list` is set to `F`. The names in `docbase_names` are the repositories that the BOCS server cannot service. However, it is recommended that you do not exclude any repositories from servicing by a BOCS server.

Asynchronous write configuration

You can enable a BOCS server for asynchronous write operations. If enabled and if a user or application chooses to use an asynchronous write, the content file is written to the repository storage area at a later time. Whether a content file is stored on a BOCS server is recorded in the `i_parked_state` property of the associated content object. Applications can use the `IDfContentAvailability.getContentAvailability` method to determine whether a content file is available in the storage area. There is also a method available, `isContentTransferCapabilityEnabled`, that indicates what content transfer capabilities are enabled in the repository.

DMS servers

The DMS server configuration is recorded in a DMS configuration object and a `dms.properties` file. The DMS configuration object is created manually after installing DMS. The properties in the DMS configuration object enable or disable the DMS server and define the URLs to be used to communicate with the DMS server. Create the DMS configuration object in the global registry repository. Create and modify DMS configuration objects only through Documentum Administrator.

The `dms.properties` file is created when DMS is installed. Most of the keys in the file are set during installation and are not modifiable. Modify those keys that are modifiable through Documentum Administrator.

Content pre-caching

You can pre-cache content on a BOCS server either programmatically or through the execution of a content pre-caching job.

To cache content programmatically, use the `IDfPrecachingOperation` package interface in DFC.

To use a job, create a pre-caching job using Documentum Administrator.

Asynchronous write

Enable asynchronous write operations in the BOCS server, in the BOCS configuration object, and in the `dm_cont_transfer_config` object. Use Documentum Administrator to enable the feature. Both the BOCS configuration object and the content transfer configuration object are stored in the global registry.

When an application or user requests an asynchronous write operation, the content file is stored, or parked, on the closest BOCS server. It is then uploaded to the repository. If it is not uploaded immediately, it is written to the repository at a later time, after an internal job, which executes the `dm_AsynchronousWrite` method, executes. That method sends the write request to DMS server again, and continues sending the message at scheduled intervals, until the content is written to the repository.

While the content is parked on the BOCS server, the `i_parked_state` property of its content object is set to `TRUE`. The content is available for reading or modification by any user who can access the BOCS server on whose host the content is parked. Other users do not have access to the content.

Asynchronous write operations affect content only. The metadata associated with the object is written to the repository as soon as the user or application requests a write operation, synchronous or asynchronous.

Note: The `dm_AynchronousWrite` job requires an index on the `i_parked_state` and `r_object_id` properties of the `dmr_content` object in the database. This index is created automatically for new repositories.

The `dm_AynchronousWrite` job is installed in the inactive state.

Communication flow descriptions

This section describes the flow of communications in a web-based environment in the following situations:

- When users request a document for viewing
- When a synchronous or asynchronous write occurs
- When a content pre-caching request occurs

Communication flow when a remote user requests a document for viewing

The following steps describe the basic communications that occur when an end user using a web browser requests a document for viewing.

1. The web application hosted on a web application server receives a request.
2. The application sends a `Getfile` request to the Content Server through the UCF facilities of the DFC on the web application server host.
3. Content Server sends back a list of the candidate content files.

For each candidate file, the list describes the following information:

- The file's location
- How far the file is from the user requesting the file
- Instructions on building the URL to access the file

Content Server digitally signs these instructions by using the private key stored in the `dm_cryptographic_key` object in the repository. The signature is used to ensure integrity and to enable authentication of the source of the instructions.

4. When the DFC receives the list of candidate content files from Content Server, it contacts a connection broker to determine the ACS servers referenced in the list that are up and running. To obtain this information, it requests a repository map from the connection broker.
5. Using the information returned by the connection broker, the DFC constructs a URL to each candidate content file that is accessible by a running ACS. The DFC sends the list of URLs to

the UCF client on the user's host machine, sorted according to their distance from the user's network location.

6. When the UCF client receives the list of URLs, it uses the first URL to request the content from the chosen ACS or BOCS server.

Depending on the configuration, the ACS server can be on a remote host machine or on the primary site's host machine.

If the UCF client receives an error using the first URL, it uses the next URL in the list it received from DFC. The UCF client reports any failed attempts to DFC, and DFC will not include the failed URLs in future requests. If all candidate URLs fail to return the content, the request fails. The client application must repeat the request; since all ACS and BOCS options failed, DFC serves the content file through the application server.

7. If the server providing the content is an ACS server, it validates the signature on the request, using the public key provided to the server. Then, it returns the content file to the UCF client. (There is one public key per repository. It is stored in the `dm_public_key` object.)

If the ACS is on a remote site and it determines that the storage areas it can access do not have the file, it sends a request to its associated Content Server. Content Server then uses surrogate get to obtain the file.

8. If the server is a BOCS server, it also validates the signature on the request, using the public key provided to the server. The server then searches its cache for the content. If the content is found, the BOCS server returns the content to the UCF client. If the content is not found, the BOCS server sends the request to the closest ACS server that has the content. The ACS server then returns the content to the BOCS server, which in turn, sends the content to the UCF client.

Note: If there are no remote ACS servers close to the BOCS server, the BOCS server requests the content from the ACS server at the primary site. To ensure faster performance, schedule frequent replication jobs to replicate content to the remote ACS server sites.

9. The UCF client writes the content to the user's local disk.

Communication flow when a remote user updates a content file

The following steps describe the basic communications that occur when an end user at a remote site updates a content file. The steps differ depending on whether a BOCS server is configured for the user's network location. It also depends on whether a BOCS server is configured for synchronous or asynchronous writing.

Asynchronous write with BOCS server

1. A user at a remote site uses a Web application to perform an operation that includes a new or updated content file.
2. The Web application uses the DFC on the web application server host to write the metadata associated with the new or update content file to the repository at the central location.
3. The Web application determines the network location of the remote user and identifies the BOCS server for that network location. Using information from the BOCS configuration object for that BOCS server, the application generates a URL for writing the content file to the BOCS

server cache. The application then passes the URL to the UCF client application running on the user's browser machine.

4. The UCF client application writes the content file to the cache of the local BOCS server, using the generated URL.
5. The web application adds an asynchronous write request to the queue for the DMS server.

From the original user's point of view, the operation is complete. The web application returns control of the browser to the user. The content file is available to all users in the same network location. However, the file is not available to users at other network locations because other BOCS servers serve these network locations. Users at other locations can view and edit the metadata, but not the content file.

6. If the BOCS server is configured for push mode, the DMS server retrieves the asynchronous write request from its queue. It then passes the write request to the BOCS server. If the BOCS server is configured for pull mode, the BOCS server contacts the DMS server and retrieves the write request.
7. The BOCS server copies the content to the ACS server. The ACS server writes a copy of the content file into a file store at the central location, without removing the copy in the BOCS cache. The content is now available to all users.

Synchronous write with BOCS server

1. A user at a remote site uses a web application to perform an operation that includes a new or updated content file.
2. The web application uses the DFC on the web application server host to write the metadata associated with the new or update content file to the repository at the central location.
3. The web application determines the network location of the remote user and identifies the BOCS server for that network location. Using information from the BOCS configuration object for that BOCS server, the application generates a URL for writing the content file to the BOCS server cache. It then passes the URL to the UCF client application running on the user's browser machine.
4. The UCF client application writes the content file to the cache of the local BOCS server, using the generated URL.
5. The web application posts a write request to the BOCS server.
6. The BOCS server copies the content to the ACS server. The ACS server writes a copy of the content file into a file store at the central location, without removing the copy in the BOCS cache. The content is now available to all users.

Synchronous write with ACS server

1. A user at a remote site uses a web application to perform an operation that includes a new or updated content file.
2. The web application writes the metadata associated with the new or update content file to the repository at the central location.

3. The web application determines the network location of the remote user. Using information from the ACS configuration object, the application generates a URL for writing the content file to the ACS server. It then passes the URL to the UCF client application running on the user's browser machine.
4. The UCF client application writes the content file to the ACS server, using the generated URL.
5. The ACS server writes a copy of the content file into a file store. The content is now available to all users.

Communication flow when a content pre-caching request occurs

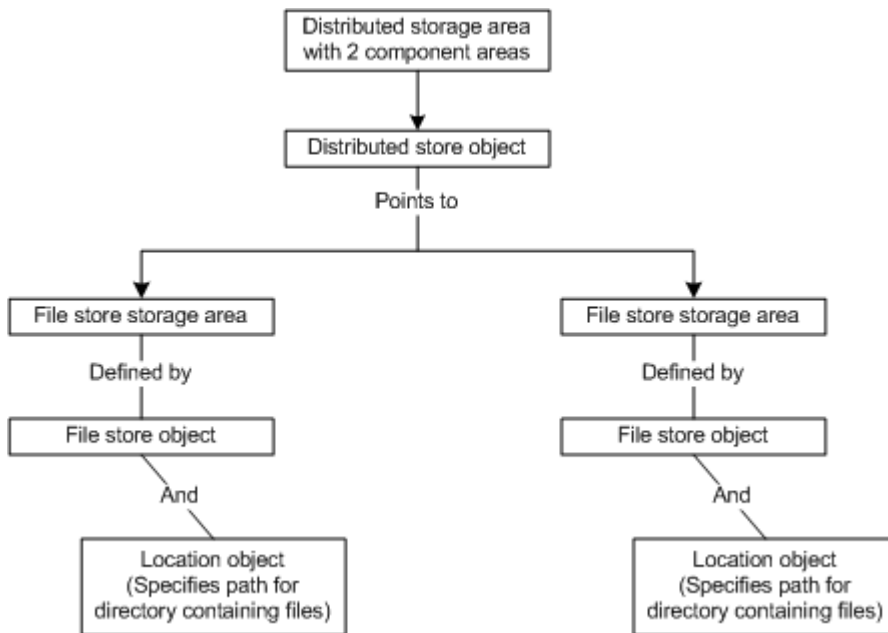
The following steps describe the basic communications that occur when a pre-caching content job runs. The communication flow is similar when an application requests pre-caching programmatically using the `IDfPrecachingOperation` package interface in DFC.

1. The method server on the Content Server host runs the pre-caching content job. The job definition identifies which objects have their content files pre-cached and to which network locations the files are pre-cached.
2. Using information from BOCS configuration objects in the global registry, the job determines which BOCS servers manage each of the selected network locations.
3. The job posts one request for each copy operation into the DMS queue. The copy operations are necessary to pre-cache the selected objects' content files to the selected BOCS caches. The BOCS servers at each remote site perform the next two steps.
4. If the BOCS server at a remote site is configured for push mode, the DMS server retrieves the pre-caching request from its queue and passes the request to the BOCS server. If the BOCS server is configured for pull mode, the BOCS server contacts the DMS server and retrieves the pre-caching request.
5. The BOCS server sends a request to the closest ACS server that has the content. The ACS server returns the content to the BOCS server, which stores it in its cache.

Implementation of distributed storage areas

A distributed store object that points to the component storage areas defines a distributed storage area. Each component storage area consists of a location and storage object.

For example, if you define a distributed storage area with two file store components, it is represented in the repository by one distributed store object, two file store objects, and the two location objects associated with the file store objects.

Figure 14. Simple example of distributed architecture

You can have as many component areas as needed. In one distributed storage area, the components can be all file stores or all linked stores, or they can be a mixture of the two types. However, all the components must have the same value in `media_type` property of their defining storage object. (The `media_type` property indicates the formats of the content files in the storage area.)

You can encrypt or compress a file store, or use content duplication checking and prevention, or use any combination of these options.

Proximity values

Proximity values are used to define:

- A Content Server's proximity to a connection broker
- An ACS server's proximity to a connection broker
- A network location's proximity to an ACS server
- Setting up multiple ACS servers for load balancing and failover

Use by Content Servers

For Content Servers, proximity values represent the server's distance from the connection brokers. The initial values are set when a Content Server is installed. You can modify the values.

The values are set in the server configuration object or in the `server.ini` file. If a server is projecting to multiple connection brokers, the proximity values sent to each connection broker must reflect the server's relative distance from each connection broker. For example, assume that a server is projecting to connection brokers, A and B, and the server is farther from connection broker B than connection

broker A. In this case, the proximity value projected to connection broker B is higher than the value projected to connection broker A. The proximity values reflects the topology of the network.

When a client requests connection information, the connection broker returns the proximity value of each known server in addition to other connection information to the client. The client uses these proximity values to distinguish between the data server and remote content servers.

To choose a server for data requests, the client looks for servers with proximity values of 0 to 999. From that group of values the client chooses the server with the lowest proximity value. If servers with a proximity value from 0 to 999 do not exist, the client looks at the servers with proximity values of 9000 to 9999. The client then chooses the server from that group with the lowest proximity value as the data server.

To choose a server for content requests, the client looks at all servers, compares the first three digits of each proximity value, and selects the lowest value. The digit in the fourth place (0 or 9) is disregarded when choosing a remote content server. Only the first three digits, counting left from the ones position, are considered. These first three digits are called a server's content proximity.

For example, assume that a server is projecting a value of 9032 to a connection broker. In this case, the server's proximity value is 9032 but its content proximity value is 032.

Use by ACS servers

When an ACS server projects to a connection broker, the information tells the connection broker that the ACS server is running. This information is used to determine which ACS server to use when handling content request for a particular user. When asked by DFC, Content Server presents list of candidate content files. The DFC also asks the connection broker for a list of the ACS servers that are running. Using both sets of information, the DFC chooses the file to return to the user.

Use by network locations

For network locations, the proximity value identifies the location's distance from an ACS server. You can record the proximity values in the ACS configuration objects representing the ACS servers that service the location. Or, the ACS server can use proximity values from its associated Content Server. DFC uses the ACS server's proximity value to determine the closest DFC to the user.

Setting up an ACS server for load balancing and failover

When you use a Documentum web client for content management and the ACS fails, the UCF client reports to the UCF server that a content transfer failure has occurred. The UCF server reports this information to the DFC, which then stops using the failed ACS until it projects again. The content is transferred through the application server until ACS starts projecting. This leads to regression in content transfer performance.

To avoid regression in content transfer performance, you can configure multiple cooperating ACSs for failover. This configuration requires multiple network locations and multiple ACSs. These ACSs can either be in different geographic locations or in the same location. In either case, to set up the ACSs for load balancing and failover, you specify different network locations and different proximity values for each ACS. When one ACS fails, another ACS can process the failed ACS's traffic. Ensure

that all the cooperating ACSs are serving a common docbase and are allowed to read/write to the filestore where content is stored. Once configured, the UCF client receives URLs corresponding to all cooperating ACSs in the order of configured network proximity values. The UCF client attempts to transfer content through all URLs in the order mentioned above until one content transfer succeeds. If content transfer through all URLs fail, the system falls back to content transfer through the application server which may or may not be seamless to the end user, depending on the operations in progress.

Note: Failover or load balancing is orchestrated by the client. If you are using a Documentum web client that does not use a UCF client for content transfer, this functionality may not exist. There is no dedicated failover or secondary ACS to another ACS. High availability deployment of ACS is also not supported.

Use the instructions in the Documentum Administrator online help to define network locations and proximity values for the ACS servers. If the ACS servers are in different locations, users use the ACS server located closest (with the lowest proximity value) to their network location. For example, users located in network A with ACS server 1 can use network A with ACS server 1. Users in network B with ACS server 2 can use network B with ACS server 2. If ACS 1 fails for an user of network A, UCF client can use ACS 2 for content transfer if it also can serve that content. Similarly, if ACS 2 fails for an user of network B, UCF client can use ACS 1 for content transfer if it also can serve that content.

Shared content

Shared content is managed through shared directories. All sites must share the distributed storage area component's directory at each site.

The feature also uses the settings of the `far_stores` property and the `only_fetch_close` property to ensure that content file reads and writes are handled correctly.

The `far_stores` property is a repeating property. It is defined for the server configuration object type, that contains the names of all storage areas that are considered far for the server. A server cannot save files into a far storage area.

For example, assume that a distributed storage area has component storage areas in Nice, Toronto, and Los Angeles. If the component storage areas in Toronto and Los Angeles are defined as far for the server in Nice, that server can only save files into the component storage area at its own site, Nice. Similarly, for the server in Toronto, if the component storage areas in Nice and Los Angeles are defined as far, the server can only save files into the component area at its site, Toronto.

The `only_fetch_close` property is defined for the distributed store object type. This property is `FALSE` by default, which allows servers to read from far storage areas. (Setting the property to `TRUE` means that servers can only read from storage areas not named in the `far_stores` property of their server configuration objects.)

Content replication

Content replication is used when a repository has a distributed storage area. There are several ways to replicate content to components of the distributed storage area:

- ContentReplication tool

The ContentReplication tool provides automatic replication on a regular schedule. This tool is implemented as a job. After you define the parameters of the job, the agent exec process executes it automatically on the schedule you defined.

- **Surrogate get**

The surrogate get feature provides replication on demand. If you use surrogate get, when users request a content file that is not in their local storage area, the server automatically searches for the file in the component storage areas and replicates it into the user's local storage area.

For surrogate get, you can use the system-defined SurrogateGet method or you can write your own program.

- **REPLICATE or IMPORT_REPLICA**

The two administration methods, REPLICATE and IMPORT_REPLICA, let you replicate content manually. On UNIX, these two methods require that the servers be able to connect using NFS.

Building block architectures for multirepository models

This section describes the features that implement a multirepository distributed object environment.

Reference links

This section describes the object types that support reference links and how reference links are handled in cross-repository applications.

Object type implementation

A reference link in a repository represents an object in another repository. Each reference link comprises a pair of objects: a dm_reference object and either a mirror object or a replica object.

Mirror objects

A mirror object is an object in one repository that mirrors an object in another repository. The term mirror object describes the object's function. It is not a type name. For example, if a user logs on to repository A and checks out a document in repository B, the server creates a document in repository A that mirrors the document in repository B. The mirror object in repository A is an object of type dm_document.

Mirror objects only include the original object's property data. Any content associated with a remote object is not copied to the local repository when a mirror object is created.

Only a limited number of operations can affect mirror objects. For example, users can link mirror objects to local folders or cabinet or retrieve their property values. Most operations affect the remote objects represented by the mirror object.

Replica objects

Replica objects are created with an object replication job is run to replicate objects from one repository to another. The term replica object represents the object's function. It is not an object type name. When an object replication job runs, it creates replicas in the target repository of the objects it is replicating. The replicas have the same object type as the source objects that were replicated.

Reference objects

Every mirror object has an associated dm_reference object. A dm_reference object is the internal link between the mirror object and the source object in the remote repository. Reference objects are persistent. They are stored in the same repository as the mirror object, and Content Server manages them. Users never see reference objects.

Valid object types for reference links

DFC supports reference links only for objects of the following types:

<ul style="list-style-type: none">• cabinet	<ul style="list-style-type: none">• query
<ul style="list-style-type: none">• docbase config	<ul style="list-style-type: none">• script
<ul style="list-style-type: none">• document	<ul style="list-style-type: none">• server config
<ul style="list-style-type: none">• folder	<ul style="list-style-type: none">• smart list
<ul style="list-style-type: none">• note	<ul style="list-style-type: none">• SysObject
<ul style="list-style-type: none">• procedure	

DFC does not support creating reference links for other object types.

Reference link binding

By default, the operation that creates the reference link also defines which version of the source object is bound to the reference link. For example, when users check out a document, they identify which version they want. If they are checking out a remote document, the specified version is bound to the reference link.

If the version is not identified as part of the operation, the server automatically binds the CURRENT version of the source object to the reference link.

You can change the default binding.

Reference link storage

When linking operation creates a reference link, the mirror object is stored in the requested location. For example, if a user links a document from repository B to Folder A in repository A, the mirror object is stored in Folder A.

When other operations create a reference link, the mirror objects are stored in folders in the local repository under /System/Distributed References. For example, mirror objects created when users check out remote objects are stored in /System/Distributed References/Checkout.

Type-specific behavior of reference links

DFC does not allow users to link local objects to a cabinet or folder that is a reference link. For example, suppose a folder from the Marketing repository is linked to a cabinet in the Engineering repository. Users in the Engineering repository cannot add objects to the folder.

If users run scripts or procedures that are reference links, DFC fetches them from the remote repository. It then runs them against the local repository session. The scripts or procedures are not run against the remote repository.

Reference link updates

A reference link is most useful when its mirror or replica object reflects the actual source object. When you change the source object, reflect those changes in the mirror or replica object. Keeping the reference link synchronized with the source occurs automatically in most cases.

The `dm_DistOperations` job automatically refreshes replicas.

Mirror objects are refreshed using an `IDfPersistentObject.refresh` method.

Operations on replica objects

When an object is replicated into a repository, the replica is given a local object ID. The operation also creates a reference object for the replica that points to the replica's source object.

When users perform operations on a replica object, the operation can affect the replica or it can affect the source object. Which is affected depends on the operation and how the method's redirection policy is set. Each DFC method that can operate on a replica has a redirection policy defined for it in an annotation. That policy determines whether the method operates on the replica or is redirected to the source object.

The redirection policies, which the `com.documentum.fc.client.distributed.replica.ReplicaRedirectPolicy` class defines, are:

- NEVER

This policy is always run against the replica object. This policy is the default behavior and is appropriate for methods that access only local properties or are always called by methods whose redirection policy has been set.

- ALWAYS

This policy redirects the method to the source object. This policy is used for methods that change global properties.

- **ON_GLOBAL_CHANGE**

This policy directs the method against the source if a global property for the replica has been changed. This policy is used for operations that read global properties.

- **RUNTIME_TEST**

This policy redirects the method to the source object based on the results of a specified runtime test.

- **ON_GLOBAL_CHANGE_AND_RUNTIME_TEST**

This policy redirects the method to the source object if a global property for the replica has been changed and a runtime test indicates that redirecting the method is appropriate. This policy is for internal use only by EMC Documentum.

The annotations are implemented using the `com.documentum.fc.client.distributed.replica.ReplicaMethodBehavior` annotation.

To perform an operation on a replica that affects the source object, the repository in which the source object resides must be active. That is, the source repository must have a server that is running.

You cannot assign replica objects to a retention policy.

Reference link security

This section describes how security is applied to reference links.

Mirror objects

Content Server for the source repository controls security for the source object, regardless whether the update is made directly to the source or through a reference link.

Content Server for the repository that contains the mirror object controls security on the mirror objects. The ACL applied to the mirror object is derived from source object's ACL using the following rules:

- If the source object's ACL is a private ACL, the server creates a copy of the ACL and assigns the copy to the mirror object.
- If the source object's ACL is a system ACL and a system ACL is in the local repository with the same name, the server assigns the local system ACL to the mirror object.

In this situation, the access control entries in the system ACLs are not required to match. Only the ACL names are matched. It is possible in these cases for users to have different permissions on the source object and the reference link.

- If the source object's ACL is a system ACL and no system ACL with a matching name is in the local repository, the server creates a local copy of the ACL. It then assigns the copy to the mirror object. The copy is a private ACL.

Replicas

The local Content Server manages replica security. Each replica is assigned an ACL when the object replication job creates the replica. The job's replication mode determines how the ACL is selected.

Object replication

Content Server's replication services allow you to replicate objects (property data and content) between repositories. Replication is automated, using jobs. Using parameters you define, the jobs dump a set of objects from one repository, called the source repository, and load them into another repository, called the target repository. You can initiate jobs by:

- The source repository
- The target repository
- A third repository that starts the job but is neither the source nor the target repository.

A repository that performs this function is called a **mediator** repository.

A replicated object in the target repository is called a replica and is designated by a replica icon.

Replication jobs

A replication job is defined as a job object in the repository that initiates the job. The properties of the job object define the parameters of the replication job. They include how often the job is executed, the source and target repositories, what is replicated, and the security assigned to the replicas. Additional properties provide status information about the job. They can tell you the time at which it last executed, the time of the next expected execution, and whether the job is active or suspended.

Jobs are created using Documentum Administrator by users with superuser privileges.

Multi-dump file replication jobs

A replication job puts the objects to be replicated in a dump file. That file is then used to load the target repository. By default, all objects replicated by a single job are placed in a single dump file. If the job replicates a large number of objects, the size of the generated dump file can be large because replication jobs replicate the specified objects and associated objects as needed.

A large dump file can be problematic. For example, the size can be a problem if the job must transfer the file to a target repository over an unreliable network. Size can also be a problem when either the source or target repository has limited disk space for the file.

To avoid problems arising from a large dump file, you can use the `-objects_per_transfer` argument to direct the replication method to break down the job into a series of smaller dump and load operations. Each of the smaller operations dumps and loads a specified portion of the objects to be replicated. For example, suppose you want to replicate 100,000 objects in a single job. You can define the job to use a single dump file, containing all 100,000 objects and associated objects. Or you could set the argument to 10,000, which causes the job to use a series of smaller dump and load operations, each replicating 10,000 objects (plus associated objects).

Replication jobs defined as a series of smaller dump and load operations continue until all objects are replicated. The job dumps a specified number of objects and associated objects, loads those objects into the target. It then dumps the next set of objects and loads those objects. No manual intervention is necessary.

Best use of multiple dump file replication

Defining a replication job to use multiple dump and load operations is beneficial if:

- If all objects are put into a single file, then the disk space on the source or target sites cannot accommodate a dump file of the size that would be generated.
- You want to avoid long, continuous remote dump or load operations
Such operations require an operational network connection for the duration of the operation, which can require special configuration of firewalls.
- An unreliable network makes it difficult to support an automatic transfer of large files.
- If the replication job encounters an error, then you want to limit the amount of lost work.

Conditions of use

Defining a replication job to use multiple dump and load operations is subject to the following conditions:

- You must define the job as a fast replication job.
- You cannot define the job as a manual transfer job.

Note: If any of the preceding conditions are true, the `-objects_per_transfer` argument is automatically set to unlimited, so that all objects are replicated in one operation.

Replication modes

There are two replication modes: nonfederated and federated. The modes determine how the `owner_name`, `acl_name`, `acl_domain`, and `a_storage_type` properties of each replicated object are handled.

Nonfederated replication mode

In the nonfederated replication mode, the `owner_name`, `acl_name`, `acl_domain`, and `a_storage_type` properties are considered local properties. This mode is typically used when the source and target repositories are not members of the same federation.

Nonfederated replication mode provides two choices for handling the `owner_name`, `acl_name`, and `acl_domain` properties in the target repository. You can:

- Remap the properties to default values provided in the replication job's definition

If you choose this option, all replicas that the job creates have the same ACL. They are stored in the same storage area in the target repository.

The ACL entries can be as open or restrictive as needed. This means that a user who has access to a document in one repository may not necessarily have access to the document's replicas in other repositories. The ACL assigned to the replica in that repository controls access in each repository.

If you choose this option and fail to provide a default ACL, the server creates a default ACL. This ACL gives Relate permission to the world and group levels and Delete permission to the owner.

- Preserve current security and storage settings when possible

If you choose this option, the `owner_name` property is reset to the default value. However, the server does not automatically assign default values to the security and storage properties. Instead:

- If the source object has a public ACL, then the server looks for a matching ACL in the target repository.

An ACL is a match if the `acl_name` and `acl_domain` values match those values of the source object's ACL. If the server finds a matching ACL, that ACL is assigned to the replica.

Otherwise, the replica's `acl_name` and `acl_domain` properties are assigned the default values specified in the job definition.

- The server looks for a matching storage area in the target repository.

If the storage area has the same name as the storage area defined in the source object's `a_storage_type` property, then the storage area is a match. If the server does not find a matching storage area in the target repository, the `a_storage_type` property in the replica is set to the storage area specified in the job definition.

Note: When a repository is configured, a storage area named `replica_filestore_01` is created. This area can be specified in the job definition as the storage area.

If there is no storage area whose names matches the source storage area and a storage area is not defined in the job, the content is placed in the default storage area defined for the `dm_sysobject` type.

- The `group_name` property of the replica object is always reset to the default group of the user specified in the replica's `owner_name` property.

Federated replication mode

In federated mode, global security controls the replica objects. The source repository retains control of the ownership and security of the replicas. The `owner_name`, `group_name`, `acl_name`, and `acl_domain` properties are not mapped to local values in the target repository. Their values can only be changed from the source repository.

This means that the `owner_name` and `group_name` properties in the replicas can contain values representing users or groups that do not exist in the target repository.

Federated replication handles ACLs in the following way:

- If the target repository contains an ACL that matches the ACL of a source object, the object's replica is assigned the local matching ACL.

An ACL is a match if its `acl_name` and `acl_domain` property values match those values of the source object's ACL. The entries in the ACL are not required to match those entries in the source object ACL.

If the source ACL is a public ACL, the `acl_domain` value is reset to `dbo` when it is written into the replication dump file. When it is loaded into the target repository, `dbo` is considered a match for the owner of the target repository.

- If there is no matching ACL in the target repository, you have two options. You can choose to:
 - Preserve security

The source ACLs are replicated into the target repository. The `acl_name` and `acl_domain` values are retained.

- Remap security

If the source ACL is an external ACL, the replica's `acl_domain` is reset to `dbo`. The `acl_name` is reset to the default value specified in the replication job definition.

In federated mode, the `a_storage_type` property is a local property. If the target repository contains a storage area of the same name, the replica content is placed in that storage area. If no storage area with the same name exists, the replica's `a_storage_type` property is reset to the storage area specified in the replication job definition. The replica's content is placed in that storage area.

Note: When a repository is configured, a storage area named `replica_filestore_01` is created. This area can be specified in the job definition as the storage area.

If there is no storage area whose names matches the source storage area and a storage area is not defined in the job, the content is placed in the default storage area defined for the `dm_sysobject` type.

What is replicated

The first time an object replication job runs, the job replicates all objects of type `dm_sysobject`. Its subtypes are linked to the source folder or cabinet (and any subfolders) specified in the job definition. The entire version tree for each object is replicated.

Note: It is up to you to determine which documents and folders to link to the source folder or cabinet and to ensure that the linking is done. The server does not select objects for replication.

The folder hierarchy within the source cabinet or folder is recreated in the target cabinet or folder. (The target cabinet or folder is defined in the job object.) The source cabinet or folder itself is not recreated in the target repository.

If a remote folder is linked to the source folder, the replication job replicates the reference link representing the remote folder. The job does not replicate the remote folder or any objects in the remote folder.

Additionally, the first run of a replication job also replicates certain objects that are related to the objects in the source folder or cabinet:

- All annotations attached to a replicated object
- All renditions of the replicated object
- If the replicated object is a parent in any user-defined relationships in the source repository, the child object and the associated relation and relation type object are replicated.
- If the replicated object is a virtual document, all of its components are replicated.

Note: If a component is a remote object and a virtual document or a folder, the replication does not traverse the remote object to replicate its children (in the case of a virtual document) or any objects it contains (in the case of a folder).

- If the replicated object has an associated assembly, all of the objects in the assembly are replicated.

Note: It is possible to create an assembly for a virtual document and attach the assembly to a simple document. If such a simple document is replicated, the components of the assembly attached to it are also replicated.

- If the object is attached to a lifecycle, the policy object representing the lifecycle is replicated.

Note: The replica lifecycle does not function as a lifecycle in the target repository. Documents cannot be attached to replica lifecycles.

- If a replicated object has properties that reference other SysObjects by their object IDs, those referenced SysObjects are also replicated with one exception.

The exception is retainer objects. If a replicated object is assigned to a retainer object (a retention policy), the retainer object is not replicated to the target repository. The `i_retainer_id` property in the replica is not set.

- If the replication mode is federated, any events associated with the replicated objects are also replicated.

What is replicated on subsequent runs of the job depends on whether the job is defined as fast replication or not.

If the job is not defined as fast replication, each time it runs, the job replicates all changed objects in the source folder. The job also examines all related objects. It replicates any objects that have changed, even if their parent object (that is, the SysObject to which they are related) is unchanged. The job also replicates any new objects in the source folder and their related objects.

In fast replication, except for annotations, the job does not consider related objects for replication unless their parent object is changed or has not been previously replicated. If an object in the source folder has not changed since the last replication job, it is not replicated. Only its annotations are considered for inclusion. New or changed annotations are replicated but the object's other related objects are not included in the job.

If an object in the source folder has changed, it is replicated. Changed related objects are not replicated unless they are also stored in or linked to the source folder.

A fast replication job also replicates any new objects in the source folder and their related objects.

For example, suppose you have a source folder named `Source_TargetB` with ten documents in it, including a virtual document called `MyVirtualDoc`. `MyVirtualDoc` has two direct components, `X` and `Y`, and `X` has one child itself, `X_1`. The document `X_1` is stored outside the source folder.

Now, suppose you run fast replication jobs with `Source_TargetB` as the source folder to repository B. The first time the job runs, the job replicates all the documents in `Source_TargetB` and their related objects. After the first run, one document, `Mydoc`, is added to the source folder and the document `X_1` is versioned. On the second job run, the job replicates `Mydoc` and all its related objects. However, `MyVirtualDoc` has not changed. Consequently, it is not replicated. Its component, `X_1` is not replicated either because it is not stored in or linked to the source folder.

In fast replication, if you want to replicate related objects regardless of changes in the parent objects, link the related objects to the source folder.

Aspect modules and replication

Object replication does not replicate aspects associated with a replicated object. If you have aspects associated with specific instances of an object type, create those aspects in the target repository. If you have default aspects defined for an object type and you replicate instances of that type, create the default aspects manually in the target repository. The aspects must be created in the target repository before performing the object replication.

Format objects and replication

EMC recommends that you manually ensure that all format objects for a particular format across participating repositories be the same.

If the format of a replicated SysObject is not present in the target repository, Content Server creates the format object in the target repository. However, if you later change the format object in the source, the changes are not reflected in the target repository. The changed format object is not re-created in the target the next time the job runs. To ensure that format definitions are the same across participating repositories, create and maintain them manually.

Data dictionary information and replication

With one exception, data dictionary information is not replicated for object types or properties. The exception is the `dm_policy` object type. If the replication job replicates a `dm_policy` object, the data dictionary information for the policy object type and its properties is replicated also.

In the target repository, data dictionary information for any object types or properties created by the replication process is available after the next execution of the data dictionary publishing job.

Display configuration information and replication

Display configuration information is information used by client applications to determine how to display one or more properties when they appear in a dialog box. Display configuration information is stored in the repository in scope configuration and display configuration objects. When you install Content Server, a script sets up default display configurations for properties in `dm_sysobject`, `dm_document`, and `dm_folder`.

Object replication does not replicate scope configuration and display objects associated with any object type. If you have defined display configuration information for types other than `dm_sysobject`, `dm_document`, or `dm_folder`, or have modified the information for these types, explicitly set it in the target repository.

If you have defined such information in the source repository and you want the same display configuration for the properties in the target repository, set it up explicitly in the target repository.

Client applications display configuration information. The *EMC Documentum Application Builder* documentation contains the instructions on defining such information.

Full refreshes

It is possible to run a replication job as a full refresh job. Such a run behaves as a first-time execution of the job and update all replicas, regardless of whether the source has changed. However, full refreshes do not change the `r_object_id` property of the replicas in the target repository.

Related objects storage

Those objects that are related to objects in the source folder and are replicated because of their association with the source objects are stored in folders called Related Objects in the target repository. There is one Related Objects folder for each replication job that targets the repository. The Related Objects folder is located under a job's target folder in the target repository.

How the replication process works

Documentum's replication services use the agent exec process that is part of Content Server. This process runs continuously and, at intervals, scans the repository for jobs that are in need of execution. The process uses properties in the job definition to determine whether to launch the job. These properties define how often the job is run and when.

When the agent exec program finds a job that must be run, it executes the method that runs the program associated with the job. This method is also specified in the job definition. For replication, the method is the `replicate_folder` method, which is created as part of the replication installation and configuration process. The `replicate_folder` method runs the `replicate_folder` program, which is provided as part of Content Server.

The `replicate_folder` program performs the actual work of replication. In the source repository, the program:

1. Dumps the source cabinet or folder to a file.
2. Performs delete synchronization.
3. Transfers the dump file to the target repository.
4. Filters the file.
5. Loads the filtered file into the target repository.

(Delete synchronization makes sure that any objects that have been deleted from the source repository are also deleted from the target repository.)

Federations

Federations are created using Documentum Administrator. Define a repository as the governing repository in a federation, add member repositories, and activate the jobs that keep global users, groups, and external ACLs synchronized.

Supporting architecture

Internally, federations are supported by:

- A dm_federation object in each participating repository
- A property in the associated repository configuration objects
- The globally_managed property in the user, group, and ACL object types
- The replicate_temp_store storage area

The federation object in each repository is named for the federation in which the repository is participating. The properties in a federation object list the federation's members and the governing repository. They also provide information about the federation's management, such as when each federation member was last updated. The *EMC Documentum System Object Reference* manual contains a list of the properties defined for the federation type.

In addition, each repository configuration object has a property called r_federation_name that contains the name of the federation to which the repository belongs, if any. This property is provided as insurance because it is possible for a repository to have multiple federation objects. In such instances, the server uses the federation object whose name matches that in the r_federation_name property in the repository configuration object.

When a repository becomes a member of a federation, the governing repository's name and r_federation_name value are projected to all its connection broker targets immediately. This action reflects the repository's new membership.

If the globally_managed property is set for a user, group, or ACL, that object is managed through the governing repository. Changes to globally managed objects in the governing repository are propagated to all member repositories. Documentum Administrator provides an easy tool for changing globally managed objects.

The replicate_temp_store storage area is used as a temporary storage place for the content of replication jobs. Object replication jobs create dump files that must be transferred from the source repository to the target repository. These files are held in the replicate_temp_store storage area until the replication job completes. Then Content Server removes the files.

The directory location of the replicate_temp_store storage area for a particular repository is %DOCUMENTUM%\data\replicate_temp_store\repository_id (Windows) or \$DOCUMENTUM/data/replicate_temp_store/repository_id, where repository_id is the hex value.

Jobs and methods

After a federation is created, global users, groups, and external ACLs are synchronized automatically using jobs and methods. Documentum Administrator manages any changes, additions, and deletions to users and groups using a change record file. The changes recorded in the file are propagated to all members by jobs. The Documentum Administrator manages external ACLs using replication jobs.

Documentum Administrator activates the jobs that perform these operations.

Multi-repository distributed models

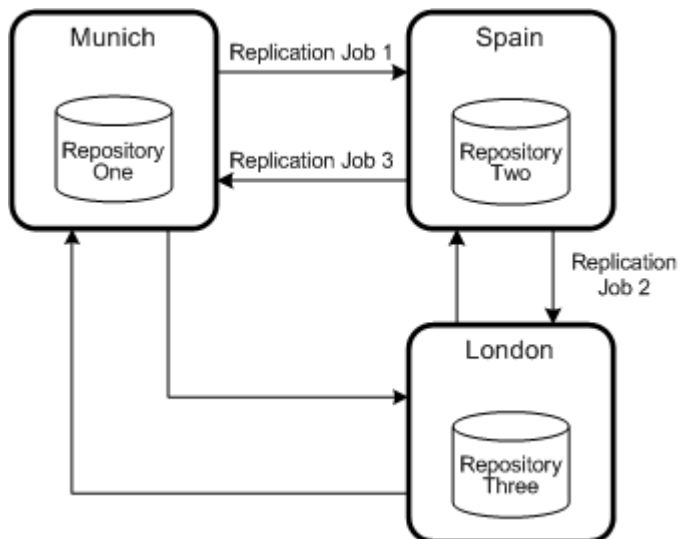
In multiple repository distributed models, entire objects, both content and metadata, are distributed between repositories. The distribution can occur through user-defined object replication jobs, or internally, when a user manipulates objects from multiple repositories in one repository session. For example, an internal replication occurs when a user starts a repository session and opens an email-attached document sent from another repository.

This section discusses the basic multi-repository replication model and the federation model. Both models are based on object replication. The federation model provides system-defined jobs that automate much of the administration work required to ensure that object replication works correctly.

Multiple repositories using object replication

Object replication replicates objects, both content and metadata, between repositories. Object replication jobs are user-defined. In object replication, there is a source and target repository. A replication job replicates objects from the source repository to the target repository. Which objects are replicated and how often the job runs is part of the job's definition. In the target repository, the replicated objects are marked as replica objects.

Figure 15. Object replication model architecture

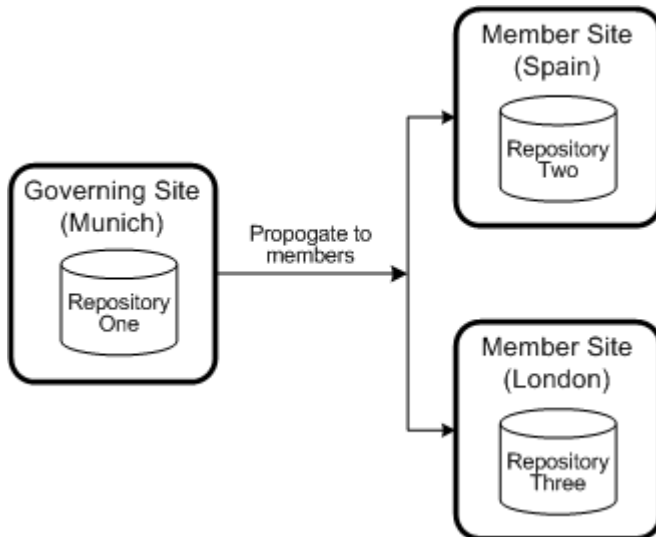


Multiple repositories working as a federation

When sites have multiple repositories and users are accessing objects from those repositories in one session, maintaining consistent security is imperative. The definitions of users, groups, and ACLs must be the same across the repositories. Keeping users, groups, and ACLs synchronized in multiple repositories can be time-consuming and error prone when the work is done manually in each repository. Placing the repositories in a federation automates much of the process.

In a federation, one repository is the governing repository. The remaining repositories are member repositories. Changes to global users and groups and external ACLs in the governing repository are propagated from the governing repository to all the member repositories automatically.

Figure 16. Federation Model



Distributed environments and the secure connection defaults

When users or applications connect to a server, they request either a secure or native (unsecured) connection. By default, all connection requests default to a request for a native connection and all servers listen on a native port. However, a server can be configured to listen on a secure port or on both a secure and a native port. If a server is listening on a secure port, then a client can request a secure connection with that server.

The building blocks that support single-repository, distributed environments rely on connections between servers within a repository. The building blocks that support multi-repository distributed environments rely on connections between repositories. Configure compatible secure connection defaults for the servers and clients to ensure successful connection requests.

There are two ways to configure the secure connection default settings to avoid connection failures:

- Set the `secure_connect_mode` property for the servers to dual.

`secure_connect_mode` is a server configuration property. If you set the property to dual for a server, the server listens on both a native and a secure port. Consequently, a client requesting a connection through that server can request either a native or a secure port with success. The Documentum Administrator online help contains complete information about the `secure_connect_mode` property and its settings.

- Set the `dfc.session.secure_connect_default` key in the client's `dfc.properties` file to either `try_native_first` or `try_secure_first`.

`try_native_first`, the default setting for this key, directs DFC to try to obtain a native connection first and if that fails, to request a secure connection. `try_secure_first` directs DFC to try to obtain a secure connection first and if that fails, to request a native connection. In either case, the target server's default setting does not affect the success of the connection request.

The jobs and features provided with Content Server that support single- or multirepository environments use a `dfc.properties` file. This file is located on the host machine on which the job resides. The `dfc.properties` file used by Documentum client products is on client application's host machine or in a network location determined that you configure. The Documentum Administrator online help provides complete information about setting the default for the client.

Distributed messaging

Distributed messaging is a feature of Content Server that supports multi-repository configurations. The server copies events from one repository to another to facilitate distributed workflows and distributed event notification.

Distributed workflow occurs when an application or user assigns an activity in a workflow definition in a repository to a user in another repository. When the activity is started, the server creates a distributed task (queue item) in the local repository that is copied to the user's home repository. It appears in the user's home repository inbox.

Distributed event notification occurs when users register for event notifications in a repository that is not their home repository. When the event occurs, the server in the repository that contains the registration creates a distributed event notification (queue item). This queue item is copied to the user's home repository. The event notification appears in the user's home repository inbox.

Distributed messaging is implemented using five properties in the `dmi_queue_item` object type. This table briefly describes these properties.

Table 3. Properties implementing distributed messaging in `dmi_queue_item`

Property	Description
<code>source_docbase</code>	The repository in which the event or task originates.
<code>target_docbase</code>	The home repository of the user receiving the event or task.
<code>remote_pending</code>	TRUE if the <code>dmi_queue_item</code> must be copied to the target repository. FALSE if it has already been copied or is not a distributed queue item.

Property	Description
source_event_id	Object ID of the source dmi_queue_item object that generated this queue item in the target repository. This property is only set in the target repository.
source_event_stamp	The i_vstamp value of the source dmi_queue_item object that generated this queue item in the target repository. This property is only set in the target repository.

Installing Documentum Messaging Services (DMS) servers

Introduction

You deploy a DMS server to an application server. You can deploy a DMS server to either a Content Server host or by itself on a different one. The DMS database is configured in the global repository's database when a new global repository is configured or old global repository is upgraded to 7.x. The DMS database consists of additional tables for storing messages. To set up high availability for DMS, you install multiple DMS servers on separate hosts (including VMs) and specify their URLs from BOCS servers.

Preinstallation requirements

Before installing DMS, the following prerequisites must be met:

- You must have created a global registry repository and the DMS server must be able to access the global registry database and the port must be open.

Note: The DMS database's user account is configured as follows:

- User name: `dms`
- Password: the same as the global repository user's password
- On the host that you want to install a DMS server, you must have already installed the Content Server binaries. You do not have to create a repository; that is, when running the Content Server installer, do not choose to install a repository.

EMC Documentum Platform and Platform Extensions Installation Guide contains more information on installing Content Server.

- On the same host as DMS server, if there exist a DB2 server or DB2 client, ensure DB2_BASE is set (DB2_BASE/java/db2jcc.jar and DB2_BASE/java/db2jcc_license_cu.jar exist). If not, you need to copy the db2jcc.jar and db2jcc_license_cu.jar jar files to DMS host. The

value of DB2_BASE could be any existing folder. Create a folder called "java" under DB2_BASE and place the two jar files.

- In an AIX with DB2 combination, dms user will not be created during the global registry configuration. You need to create an AIX operating system user named "dms" to configure DMS before launching DMS installation.
- A DMS server and content server instance must be the same version.
- A DMS server must have HTTP access to all BOCS servers in push mode (servers in push mode receive messages from the DMS server).
- To set up high availability properly, EMC recommends that you install only one DMS server per host.

Installing a DMS server

1. Start the Content Server configuration program as follows:
 - UNIX and Linux: Run `$DM_HOME/install/Server_Configuration_Program.sh` and select **Documentum Messaging Service (DMS)**.
 - Windows: Select **Start > All Programs > Documentum > Documentum Server Manager** and on the **Utilities** tab, click **Server Configuration**, and then select **Documentum Messaging Service (DMS)**.

2. Complete the installation as instructed.

EMC Documentum Platform and Platform Extensions Installation Guide contains more information on the connection broker and global registry.

Table 4. Distributed Environment Fields

Dialog Box	Field/Choice	Default	Description
DMS client - application server credentials	Admin user password	N/A	<p>The password for the <code>dmsAdmin</code> administrator of the DMS administration (JMX console) web application. The password is encrypted.</p> <p>The corresponding <code>dms.properties</code> property is: <code>dms.jmx.password</code></p>
	Host IP	N/A	<p>The IP address of the host of the DMS web application.</p> <p>The corresponding <code>dms.properties</code> property is: <code>dms.jmx.host</code></p>
	Listener Port	8489	<p>The number of the port at which the DMS web application listens. The listener port number of the DMS administration (JMX console) web application is this port number plus one (the default is 8490). The <code>dms.properties</code> property is: <code>dms.rmi.registry.port</code></p>

Dialog Box	Field/Choice	Default	Description
DMS client - connection broker	Host name	N/A	Name of the host of the connection broker. The connection broker must be able to connect to the global registry repository.
	Host port	1489 (native) 1490 (SSL)	Port number of the connection broker.
	Use certificates	N/A	<p>Select Use certificates if Content server and global repository is already configured with SSL certificates.</p> <p>Provide the DFC trust store information:</p> <ul style="list-style-type: none"> • TrustStore: The location of the DFC trust store. For example, \$DOCUMENTUM\dba\secure\dfc.keystore • Password: The password of the trust store file. • Select Use Default Java TrustStore if you want to use the default DFC Java trust store.
DMS client - select global registries	Global registry login name	dm_bof_registry	Name of the global repository user.
	Global registry login password	N/A	Password for the global repository user.

Dialog Box	Field/Choice	Default	Description
Documentum Messaging Server - Database information	Database type	Database that you have installed.	Select a database from the list box.
	Database host name	N/A	The name of the host on which the global registry database system resides.
	Database listener port	1433 (SQL Server) 1521 (Oracle)	The port at which the global registry database system listens for requests.
	DMS user name	N/A	Global registry database owner name.
	DMS password	N/A	Global registry database owner password.
	Database name	N/A	(SQL Server only) The name of the database that was created for the global registry repository. The name of this database is specified in the <code>database_name</code> property in the <code>server.ini</code> file.
	Database instance name	N/A	Name of the instance if the database is installed with a named instance.
	Repository owner name	N/A	(Oracle only) The name of the global repository database owner. This name is used as the name of the database table. You can find the name of this database in the <code>server.ini</code> file.
	Oracle SID	N/A	(Oracle only) The Oracle system ID

In the **Summary** dialog box, the post and consume URLs are displayed. You specify these URLs to route messages to and from BOCS servers. describes the DMS default directories.

Table 5. DMS Default Directories

Directory	Description	Default Directory
Installation directory	Root directory of the DMS web application.	Windows: %DOCUMENTUM%\wildfly_directory\server\DctmServer_DMS UNIX and Linux: \$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_DMS
Properties files directory	The installation process configures dms.properties automatically. See dms-full.properties for a description of the properties in dms.properties. You use Documentum Administrator to configure a DMS server (these configuration changes are automatically made to dms.properties). log4j.properties contains logging properties for DMS.	Windows: %DOCUMENTUM%\wildfly_directory\server\DctmServer_DMS\deploy\DMS.ear\lib\configs.jar UNIX and Linux: \$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_DMS/deploy/DMS.ear/lib/configs.jar

3. By default, logging when messages are sent and received is disabled; otherwise, the DMS log can become very large. You might need to enable payload logging for troubleshooting purposes. To change the level at which CXF log statements are logged, add the following in the %DOCUMENTUM%\wildfly_directory\server\DctmServer_DMS\configuration\standalone.xml:

```
<subsystem xmlns="urn:wildfly:domain:logging:1.1">
  <logger category="org.apache.cxf">
    <level name="ERROR"/>
  </logger>
  <logger category="org.apache.cxf.services">
    <level name="ERROR"/>
  </logger>
</subsystem>
```

By default, the log levels are set to ERROR. This level can be set to any of the log levels used by log4j.

Configuring a DMS server

You use Documentum Administrator to:

- Add and delete DMS server configurations for a repository.
- Enable and disable messaging.
- Configure BOCS server message routing.

Note: Defaults are set for the timing of the pushing of messages to BOCS servers. Change the timing in `dms.properties`. You cannot use Documentum Administrator to change the timing.

- Access the resource agent. The file is administered using Java Management Extensions (JMX), accessed through the DMS resource agent in Documentum Administrator. The resource agent is accessed through the Resource Management node. Documentum Administrator online help or the *EMC Documentum Content Server Administration and Configuration Guide* contains the instructions on accessing the resource agent. The URL for the JMX resource agent is as follows:

```
service:jmx:rmi:///jndi/rmi://host_name:port/dms
```

where:

- `host_name` is the name or IP address for the host computer.
- `port` is the assigned port. The default port is 8490.

If you want to connect to the DMS administration resource agent from outside of a firewall, configure the firewall settings to allow the RMI protocol for the port.

Timing the pushing of messages to BOCS servers

If a BOCS server is unreachable when messages are pushed to it, then the DMS server waits for a specific amount of time before resending messages. The `dms.properties` file's `dms.job.push.scheduler.delay.milliseconds` property specifies this wait time. Its default is 10 minutes.

DMS pushes messages to a BOCS server at regular intervals. The `dms.properties` file's `dms.job.push.scheduler.delay.milliseconds` property specifies this interval. Its default is 1 minute.

Best practices

Quality improvements have been made in the area of DMS 7.0 high-availability (HA). For example, the CXF framework is used to deliver messages. The CXF framework is faster, simpler, and more scalable than prior methods used in delivering messages. To make DMS and DMS HA perform better, apply the best practices described in the following sections.

Note: The best practices and/or test results are derived or obtained after testing the product in the EMC testing environment. Every effort is made to simulate common customer usage scenarios during performance testing, but actual performance results will vary due to differences in hardware and software configurations, data, and other variables.

Database settings for the Oracle-based DMS database

- For the DMS database, follow the general database guidelines provided in the *EMC Documentum Content Server Administration and Configuration Guide*.
- For large or rapid volumes of asynchronous writes and pre-caching activities, EMC recommends that you monitor the database logs and statistics reports, so that the database does not become a DMS performance bottleneck.

DMS configuration

- The performance of DMS/BOCS message processing is dependent on the number of messages and is not related to the size of the cached data.
- For information about settings related to performance, see `dms-full.properties`.
- For rapid ingestions, set the push delay to 1 second. This setting can help push messages to BOCS as soon as possible. For example:

```
dms.job.push.scheduler.delay.milliseconds = 1000
```

However, you can also set the delay for longer periods such as 1 minutes or 2 minutes because the delivery is not sensitive to time.

- For a single DMS server, the maximum delivery rate can reach 50 – 70 messages per second when no messages are inserted. If messages are inserted, the delivery rate can drop to 25 message per second.
- To increase the delivery rate, adjust the following properties according to the `dms-full.properties` file:

```
# queue.size must be equal to or more than
dms.message.fetch.batch.size
# task must be less than queue size,
this is parallel task that consume the queue data.
dms.message.fetch.batch.size=100
#The task-size is the number of thread
dms.job.push.queue.size=100
dms.job.push.task.size=10
```

When a DMS server is started, it creates a number of threads based on the `dms.push.task.count` parameter for pushing the messages to BOCS. However, one has to be careful in changing these values. For instance, setting the batch size to a higher value can prolong the database lock that is obtained for processing each batch.

- To improve DMS message processing, the expired messages must be cleaned up from the database periodically. To delete messages, one can use the following queries:

```
—
delete from dms_destination_message where processed_date
<time_in_milliseconds(currentdate - 180) and status_no = 2
—

delete from dms_message where processed_date
<time_in_milliseconds(currentdate - 180) and status_no = 2 and
m.message_id in ( select message_id from dms_destination_message
where processed_date < time_in_milliseconds(currentdate - 180)
and status_no = 2)
```

- When pre-caching or async write jobs are scheduled very frequently and/or concurrently, DMS can become a bottleneck because a flood of message queues can come in at a faster rate (if there are not enough DMS nodes and BOCS servers to handle the load). If possible, run these jobs at times of low user activity.
- By default, only the primary node in DMS HA configuration can receive messages from DMS clients such as JMS or BOCS. When the primary node goes down, activities fail over without loss of messages from the primary node to other nodes. However, in DMS HA, multiple nodes can be active in delivering messages to BOCS, which increases the message-delivery throughput to BOCS.

BOCS configuration

- To allow the transfer of a high volume of documents, set the `cache.paging.page_size` or the `cache.paging.max_count` parameter to a large value. For example:
`cache.paging.page_size=10000`
- A BOCS server has limitations when processing very large volumes of requests for content transfer. The unavailability of BOCS can make content failover to ACS, transparent to the user. Even 10 requests per second can cause such a failover to ACS. Therefore, EMC recommends adding more BOCS servers in rapid ingestion scenarios.
- To improve BOCS processing messages, you can increase the number of threads that write parked content to ACS (if the server can create request threads). The default value of the `bocs.asynch.executor.max.thread.count` parameter is 2.

Starting and stopping a DMS server

Use these instructions to start and stop the DMS server, the database, and the application server.

Table 6. Starting and Stopping DMS Servers

Operating System	Starting DMS	Stopping DMS
Windows	In the Services dialog box, right-click Documentum Messaging Server and select Start .	In the Services dialog box, right-click Documentum Messaging Server and select Stop .
UNIX and Linux	Run <code>\$DOCUMENTUM_SHARED/wildfly_directory/server/startDMS.sh</code>	Run <code>\$DOCUMENTUM_SHARED/wildfly_directory/server/stopDMS.sh</code>

Uninstalling a DMS server

Make sure to remove the corresponding DMS server configuration from all repositories.

To remove a DMS server:

- (Windows only) In the **Services** dialog box, right-click **Documentum Messaging Server** and select **Stop**.
- Start the Content Server configuration program as follows:
 - UNIX and Linux: Run `$DM_HOME/install/dm_launch_server_config_program.sh` and then select **Documentum Messaging Service (DMS)**.
 - Windows: Select **Start > All Programs > Documentum > Documentum Server Manager** and on the **Utilities** tab, click **Server Configuration**, and then select **Documentum Messaging Service (DMS)**.
- Select **Delete DMS instance**.

Installing BOCS

Overview

This section contains the instructions for installing, removing, and starting and stopping the product.

Note: The BOCS server is one component of a distributed configuration. *EMC Documentum Platform and Platform Extensions Installation Guide* contains the information on installing other components, such as Content Server.

The BOCS server is supported on the same platforms as Content Server and you can install in a heterogeneous environment. The database client is not required on the BOCS server host because the BOCS server does not interact with the database.

BOCS server environment

BOCS servers are supported in heterogeneous environments. For example, a BOCS server installed on a Windows host might serve a repository and Content Servers running on UNIX or Linux, or a BOCS server installed on a UNIX or Linux host might serve a repository and Content Server on a Windows host.

Preinstallation requirements

Before installing BOCS, ensure that the following prerequisites are met:

- The host environment meets the system requirements.
- If you do not want to accept the defaults, choose directories and port numbers available for use during installation

Default directory locations are suggested for the installation directory, the content cache, and the parked content cache. Default port numbers are suggested for the BOCS instances (8086 for the instance server).

- You know the host name of the DMS installation and port number to use to communicate with the DMS server (if the BOCS server is configured to operate in pull mode).
- Choose the user name and password for the user who administers the BOCS server's `acs.properties` file through the resource agent in Documentum Administrator.
- If you are installing on UNIX and Linux, ensure that the `DOCUMENTUM` and `DOCUMENTUM_SHARED` environment variables are set.

Set `$DOCUMENTUM` to the installation directory.

Set `$DOCUMENTUM_SHARED` to the directory in which you want to install DFC on the BOCS host. This directory is also the top-level directory under which the scripts used to remove the components are stored.

Note: The installation owner must have read, write, and execute permissions on these directories and their subdirectories.

- A password for the bocsAdmin user is chosen.

The bocsAdmin user is the user who administers the BOCS server's acs.properties file through the resource agent in Documentum Administrator.

Installing BOCS

Download and extract the compressed distribution file.

1. Run the installation program.

- Windows: bocsSetup.exe
- UNIX and Linux: bocsSetup.bin

If an error occurs during the setup for installation, that error is recorded in a setuperror.log file. During the installation process, any messages relating to the process are saved to a file named install.log. Both of these files are stored in the same directory where the downloaded files were decompressed.

2. Accept the terms of the license agreement and click **Next**.
3. Specify the destination directory. For example, C:\Documentum and click **Next**.
4. Type the installation owner's password and click **Next**.
5. Type the administrative user's password and click **Next**. The default port is 8086.
6. On the **Specify BOCS options** dialog box, specify the location and maximum sizes of the content caches:

- To choose a directory for the cache directory that is different from the default, use **Browse**
- To choose a maximum size for the cache directory that is different from the default, enter the new value (in megabytes).
- To choose a directory for the Prime Store directory that is different from the default, use **Browse**.

The Prime Store directory is the location where parked content is held.

- To choose a maximum size for the Prime Store directory that is different from the default, enter the new value (in megabytes).
7. On the **Pull Mode** dialog box, enable or disable pull mode as follows:
 - a. To enable push mode, deselect **Enable Pull Mode**.
 - b. To enable pull mode, select **Enable Pull Mode**.
 - c. In the **DMS URL** field, enter the URL that the BOCS server uses to connect to DMS. If you are connecting to multiple DMS servers in HA mode, enter multiple, comma-separated DMS URLs.

In pull mode, BOCS servers retrieve messages from DMS. The URL must be in the following format:

`http://host:port`

where *host* is the host of the DMS server and *port* is the port on which the DMS server is listening.

- d. In the **Config Object Name** field, enter the name of the BOCS configuration object.
Record the name you select for later use when actually creating the BOCS configuration object. The name is recorded in the `acs.properties` file by the installation program. If you use a different name when creating the BOCS configuration object, edit the `acs.properties` file to change the name.
- e. Record the **Path to Certificate** value that is displayed.
This directory path is required later, when the BOCS configuration object is created.
8. In the **Proxy Settings** dialog box, indicate whether a proxy server is to be installed between BOCS and Content Server.
 - a. If you do not want to install a proxy server, select **No**.
 - b. If you want to install a proxy server, select **Yes** and specify the proxy server's host name and the port used to contact the proxy server.
9. Click **Install**.
10. Using Documentum Administrator, create a BOCS configuration object in the global registry.

Note: Navigate to `$DOCUMENTUM\patch` and open the `patch-info.xml` to view the product/version details. The `patch-info.xml` file displays the installed product version (base or patch or both). It is not meant for patch release versions only.

Configuring BOCS

This section describes how to configure BOCS.

Configuration requirements

To use BOCS servers:

- A BOCS configuration object representing the BOCS server must reside in the global registry.
- Configure an `acs.properties` file correctly for each BOCS server. The installation process configures an `acs.properties` file automatically.

More information on the `acs.properties` file on the BOCS server host is provided in the section and the sections that follow.

- The ports on which BOCS server serves content to users must be open.
- If the BOCS server is configured in pull mode, the BOCS server must have access to the URL defined in the `dms.pulling.url` property in the `acs.properties` file.
- If the BOCS server is configured in push mode, the DMS server must have HTTP access to the BOCS server.
- To use the pre-caching feature, the capability must be enabled in the content transfer configuration object in a repository that owns the content.

Note: This feature is enabled by default.

- If BOCS is configured to use `Asynchronous Write` mode, ensure that the job `dm_AsyncWrite` is enabled in the Content Server. Use Documentum Administrator to enable the `dm_AsyncWrite` job.

DMS and global registry compatibility requirements

Pre-caching jobs and asynchronous content transfers require DFC to send certain information to BOCS. DFC first sends this information to DMS, and DMS delivers this information to BOCS later. DFC uses the `DmsClient` service-based business object (SBO) to communicate to DMS. This SBO is deployed in a global registry repository and is downloaded by using Business Object Framework (BOF). For the `DmsClient` SBO to correspond to DMS, DMS and the global registry must be of the same version. This situation is not considered to be an issue because there is only one DMS and global registry per installation.

The `acs.properties` file

This section provides basic information about the file.

`acs.properties` file location

The `acs.properties` file is located as follows:

- Windows:

```
%DOCUMENTUM%\wildfly_directory\server\DctmServer_BOCS\deploy\
bocs.ear\lib\configs.jar\config
```

- UNIX and Linux:

```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_BOCS/deploy/
bocs.ear/lib/configs.jar/config
```

Note: The `acsfull.properties` file, which is in the same directory as `acs.properties` is located, describes all parameters and their default values.

File administration

The file is administered using Java Management Extensions (JMX), accessed through the BOCS resource agent in Documentum Administrator. The resource agent is accessed through the Resource Management node. Documentum Administrator online help or the *EMC Documentum Content Server Administration and Configuration Guide* contains the instructions on accessing the resource agent. The URL for the JMX resource agent is as follows:

```
service:jmx:rmi:///jndi/rmi://host_name:port/bocs
```


where:

- *host_name* is the name or IP address for the host computer.
- *port* is the assigned port. The default port is 8087.

If you want to connect to the BOCS administration resource agent from outside of a firewall, configure the firewall settings to allow the RMI protocol for the port.

Changing the JMX user password

When you installed BOCS, you provided a user name and password for the JMX server. You can change the password for that user.

To change the JMX user password:

1. Log on to Documentum Administrator as the installation owner.
2. Navigate to the BOCS resource agent.
3. Double-click the resource agent to display the Mbeans in the agent.
4. Right-click the JmxUserManagementMBean and click **Operations**.
5. Click **changePassword**.
6. Type the current user name and the new password.
7. Click **Start Operation**.

Default settings

The BOCS installation program explicitly sets a number of configuration keys in the `acs.properties` file for the BOCS server.

Configuration keys that cannot be changed

Do not change the `mode.cachestoreonly=false` configuration key, which the installation program sets.

The following keys define the server associated with the file as a BOCS server.

- `jms.url`
- `java.naming.factory.url.pkgs`
- `jms.connection.factory`
- `jms.queue.name`
- `jndi.factory`

Configuration keys that to be modified only as a part of BOCS reconfiguration

Modify the following keys only as part of BOCS reconfiguration:

- `bocs.keystore`
- `bocs.configuration.name`
- `bocs.pulling.mode.enabled`
- `dms.pulling.url`
- `dms.server.base.urls`
- `bocs.pulling.interval`
- `proxy.host`
- `proxy.port`
- `policy`
- `policy.1`

Configuration keys that you can change

You can change the following configuration keys that the installation program sets:

- `cache.store.root`
This key identifies the root cache directory. The BOCS server stores cached content in this directory. The default name of the directory created by the installation program is `acsCache`.
- `cache.store.quota`
This key specifies the size of the root cache directory. It is set to 1 GB by default.
- `primary.content.store.root`
- `primary.content.store.quota`
- `tracing.enabled`
- `mode.debug`

Adding or modifying a root cache directory

The BOCS server caches content requested by users in the root cache directories. Root cache directories are identified in the `cache.store.root` key or keys in the file.

You can add additional root cache directory locations or change the size of a root cache.

The key that identifies a root cache directory is:

```
cache.store.root
```

The key that defines the size of a root cache directory is:

```
cache.store.quota
```

If you add additional cache directories, be sure to define their sizes also. The size specified in a `cache.store.quota` key is applied to the directory identified in the matching `cache.store.root` key. For example, if you add a directory specified in key `cache.store.root.1`, the size of that directory is set in `cache.store.quota.1`.

Defining the parked content directory

If the BOCS server uses asynchronous write operations, define the location and maximum size of the directory that stores the parked content. This directory is not configured by default when the BOCS server is installed.

To define the directory's location set the following key in the BOCS server's `acs.properties` file:

```
primary.content.store.root=fully_qualified_path
```

where *fully_qualified_path* is the path to the directory location where you want to store parked content. This directory must not be a subdirectory of the directory specified in the `cache.store.root` key.

The maximum size of the directory is defined in the following key:

```
primary.content.store.quota
```

Configuring cache housekeeping

The housekeeper runs once a day by default. Additionally, if the amount of content in the cache reaches 80% of the cache's configured capacity, the housekeeper runs outside of the scheduled execution to purge content to free space in the cache.

There are several keys in the `acs.properties` file that control housekeeping for the BOCS server. The values in these keys are applied to the content in all cache root directories. The following table describes the keys.

Table 7. `acs.properties` keys controlling BOCS cache housekeeping

Parameter	Description
<code>cache.schedule.housekeeper</code>	Controls how often the housekeeping process runs by default to find and remove obsolete files from the cache. The value is interpreted as seconds. The default value is 24*3600 (once a day).

Parameter	Description
cache.schedule.housekeeper.start_delay	<p>Defines the interval between the startup of the BOCS server and the first run of the housekeeper.</p> <p>The value is interpreted in seconds. The default is 5*60 (5 minutes)</p>
cache.schedule.housekeeper.retire_interval	<p>Defines how long a cached, content that has not been accessed may be in the cache before being considered obsolete.</p> <p>The value is interpreted as seconds and represents an interval counted from the last time the cached content was accessed by a user. If the specified number of seconds passes and the content has not been accessed within that interval, the housekeeper considers the content as obsolete and removes it from the cache.</p> <p>The default is 30*24*3600 (30 days)</p>

Configuring consistency checks

The system periodically validates the consistency of content in the cache. There are two keys that control consistency checks. The following table describes the keys.

Table 8. acs.properties keys controlling cache consistency checking

Parameter	Description
cache.schedule.consistency	<p>Controls how often the consistency of the cache is validated.</p> <p>The value is interpreted as seconds. The default value is 24*3600 (once a day).</p>
cache.schedule.consistency.retire_interval	<p>Defines the interval between the startup of the BOCS server and the first run consistency check.</p> <p>The value is interpreted in seconds. The default is 5*60 (5 minutes)</p>

Configuring cache write intervals

The BOCS server keeps in memory information about available on BOCS content. By default, the BOCS server saves this information on a disk at specified intervals. The default interval is every 5 minutes. To change that interval, specify a number of seconds in the following key:

```
cache.schedule.flush_directory
```

Configuring use of content retrieval URLs

There are two keys that control how a BOCS server handles the URLs used to retrieve content files. The following table describes the parameters.

Table 9. acs.properties keys controlling URL use

Parameter	Description
repository.validation.delta	<p>Defines the length of time for which a URL for content retrieval is valid. The interval is counted from the time the URL was generated.</p> <p>The value is interpreted as minutes. The default value is 360 (6 hours).</p>
validate.certificate	<p>Controls whether the BOCS server validates the public key certificate used for URL validation against the list of trusted certificates.</p> <p>The default is false, meaning that the public key certificate is not validated.</p>

Log files

The following log files are maintained in a BOCS installation:

- An application server log file
- BOCS-specific DFC log file
- A log file for all DFC log messages
- A log file recording all messages sent and received by BOCS

Application server log file

The application server log file (`server.log`) records application server errors and is located as follows

- Windows:


```
user_directory\wildfly_directory\server\DctmServer_BOCS\log
```
- UNIX and Linux:


```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_BOCS/log
```

BOCS-specific DFC log file

The BOCS-specific log file records DFC information operations related to the BOCS server.

The default name and location of the file is:

Note: The name and location of the file are defined in `log4j.properties`.

- Windows:

```
user_directory\wildfly_directory\server\DctmServer_BOCS\logs\AcsServer.log
```

- UNIX and Linux:

```
$DOCUMENTUM_SHARED/wildfly_directory/server/DctmServer_BOCS/logs/AcsServer.log
```

Log file for all DFC messages

By default, the name of the log file that records all DFC log messages is `log4j.log`. The default location of the file is:

Note: The name and location of the file are defined in the `log4j.properties` file. The default size for this file is 100 Kb.

- Windows

```
user_directory\wildfly_directory\server\DctmServer_BOCS\logs\bocs.log
```

- UNIX and Linux:

```
$DOCUMENTUM_SHARED/wildfly_directory\server\DctmServer_BOCS\logs\bocs.log
```

Log file for recording BOCS messages

The `access.log` file records information about messages sent, received, and pulled by a BOCS server. The entries in the file have the following format:

```
128.222.102.204 - - [04/Jun/2007:18:31:11 -0400]
"POST /bocs/servlet/ACS HTTP/1.1" 200 296 128.222.102.204
- - [04/Jun/2007:18:31:32 -0400]
"GET /bocs/servlet/Handshake HTTP/1.1" 200 19 128.222.102.204
- - [04/Jun/2007:18:31:32 -0400]
"POST /bocs/servlet/ACS HTTP/1.1" 200 60
```

Note: Entries are line-wrapped in the documentation only. They appear on one line in the file.

Each entry records the following:

- The IP address that originated the request
- The date on which the request was processed
- The request itself
- The return status
- The length of the reply, in bytes.

The return status is either 200, representing a good status, or 500, representing an error status.

Log file size and backups

The maximum size of the application server and DFC log files is defined in the `log4j.properties` file, in the `MaxFileSize` entry for each log file. You can set the size by setting the `MaxFileSize` key for the log file.

Log files are not backed up by default. Instead, when the file reaches its maximum size, the system overwrites the file. If you want to save backups of the log files, set the `MaxBackupIndex` key in the `log4j.properties` file for the specific file. If that is set, when a log file reaches its specified maximum file size, it is backed up and another file is started. The name of the backup file is `log_file_name.log.N`, where *N* starts with 0 and increments by 1 with each backup. For example, `AcsServer.log.0`, `AcsServer.log.1`, and so on. The first backup of the log file is numbered 0, the second is numbered 1, the third is numbered 2, and so on. This means that the lower the number on the backup file, the older the file is.

The value set in `MaxBackupIndex` determines the number of saved backup files.

Reconfiguring a push BOCS server to pull mode

Use the information in this section to change a BOCS server running in push mode to run in pull mode.

To reconfigure a BOCS on push mode to run in pull mode:

1. Log on to Documentum Administrator and connect to the global registry.
2. Access the `acs.properties` file, through the BOCS resource agent, to:

- a. Check the file to ensure that the BOCS configuration name is set:

```
bocs.configuration.name=name_of_bocs_config_object
```

The certificate required for pull mode is associated with the BOCS server through the name of the BOCS configuration object. The name in `bocs.configuration.name` must match the name identified in the certificate.

- b. Set the mode to pull.

```
bocs.pulling.mode=TRUE
```

- c. Define the DMS server's consume URLs as follows:

```
dms.pulling.url=BOCS,DMS_URL_1,DMS_URL_2 dms.server.base.urls
=BOCS, DMS_URL_1,DMS_URL_2
```

You enable DMS high-availability by specifying multiple DMS server URLs in a comma-delimited format (that is, *DMS_URL_1* and *DMS_URL_2*).

3. Execute the following script to create the BOCS certificate:

- Windows:

```
user_directory\bocs\bin\generateCert.bat certificate_file_path
```

- UNIX and Linux:

```
$DOCUMENTUM/bocs/bin/generateCert.sh certificate_file_path
```

where *certificate_file_path* is the location where you want to store the certificate.

4. Restart the BOCS server.
5. Use Documentum Administrator to modify the BOCS configuration object identified in `bocs.configuration.name` in the `acs.properties` file.

- a. Change the mode from push to pull.

This setting is on the **Security** tab of the **BOCS Server Configuration Properties** page.

- b. Upload the BOCS certificate you generated to the BOCS configuration object.
6. Make sure that the global registry connection for the DMS is correctly set.

Enabling BOCS access from behind a firewall

For enabling BOCS access from behind a firewall, the two following assumptions are made:

- BOCS is on a host named `dctm_bocs01` and the port 8086 is used.
- An Apache HTTP server is used as the reverse proxy and the host name is `proxy01`. HTTP is used as the communication protocol.

Add the following directives to `httpd.conf` of the Apache HTTP server to map `http://proxy01/bocs` to `http://dctm_bocs01:8086/bocs` and `http://proxy01/bocs-ws` to `http://dctm_bocs01:8086/bocs-ws`:

```
ProxyPass /bocs http://dctm_bocs01:8086/bocs ProxyPassReverse
/bocs http://dctm_bocs01:8086/ bocs ProxyPass
/bocs-ws http://dctm_bocs01:8086
/bocs-ws ProxyPassReverse
/bocs-ws http://dctm_bocs01:8086/bocs-ws
```

Set the BOCS URL to `http://proxy01/bocs/servlet/ACS`.

Note:

- If you access BOCS through a firewall and reverse proxy, some additional configuration is required.
- In this example, an Apache HTTP server is used and the instructions can only be used as a reference if other reverse servers are used.

Enabling or disabling the write mode

You can configure a BOCS server to use both asynchronous and synchronous write or only synchronous write or neither mode.

Use Documentum Administrator to set the properties in the BOCS configuration object and content transfer configuration object and enable messaging in DMS.

To set write modes for a BOCS server:

1. In the BOCS server properties page, modify the content access to specify the content write access you want to allow for the BOCS server.
In the repository, this information is recorded in the BOCS configuration object, in the `rw_capability` property.
2. In the Distributed Transfer Settings Properties page, set ACS Write to allow read and write capabilities.
In the repository, this information is recorded in the content transfer configuration object, in the `acs_write_mode` property.
3. In the Messaging Server Configuration Properties page, enable messaging.

DNS requirement for web-based client hosts in distributed environment

In a configuration that includes ACS or BOCS servers, the machines hosting the web browsers must be able to resolve the base URLs defined for the ACS or BOCS servers using DNS. (The base URLs for the servers are recorded in the ACS configuration object for the server. You can modify it using DA. Alternatively you can use the Documentum Query Language (DQL) to modify the base URLs after they are set).

Creating a BOCS configuration object

After you install BOCS or upgrade an existing installation, create a BOCS configuration object for the BOCS server. The Documentum Administrator online help contains the instructions on using Documentum Administrator to create the BOCS configuration object.

Configuring security for BOCS servers in pull mode

When you install or upgrade BOCS, the installation program generates a public key. A BOCS server in pull mode uses that key to generate a digital signature. It identifies itself with this signature when it contacts a DMS server to pull messages from the DMS server's queue.

If you enable pull mode during BOCS installation, the installation program provides the path to the public key certificate. When you create the BOCS configuration object, provide the file path for that key. The location of the public key is recorded in the BOCS configuration object.

Installing and configuring BOCS on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.
3. Provide all the required details in the `bocs_conf.conf` file. Read the description of every field and provide valid values for each parameter.
4. Run the `bocs.sh` script.
5. To verify the installation, check `http://<dockerbaseip>:8086/bocs/servlet/ACS`. Also, check the web application server logs. For example, the logs at `/opt/tomcat/logs`.

Upgrading BOCS

EMC Documentum System Upgrade and Migration Guide contains the detailed information.

Removing BOCS

To remove BOCS, remove the **Documentum Branch Office Caching Services** installed service.

On Windows

1. Log on to the BOCS server host as the installation owner.
2. Use the Windows **Service** to stop the BOCS server.
3. Use the **Control Panel** to remove the BOCS service.

On UNIX and Linux

1. Log on to the BOCS server host as the installation owner.
2. Run the following program:

```
$DOCUMENTUM/uninstall/bocs/Uninstall
```

Starting and stopping BOCS

On Windows

BOCS servers are installed as Windows services. To start and stop BOCS servers, use the Services dialog box. The service name is Documentum Branch Office Caching Services.

On UNIX and Linux

The internal database must be running before you start BOCS. However, if the database is not running, no error is generated. Use scripts to start and stop BOCS and to start and stop the internal database. If you stop BOCS, you do not need to stop the internal database.

To start BOCS:

Run `$DOCUMENTUM/wildfly_directory/server/startBOCS.sh`.

To stop BOCS:

Run `$DOCUMENTUM/wildfly_directory/server/stopBOCS.sh`.

Configuring WildFly, ACS, BOCS, and DMS for Secure Socket Layer (SSL) connections

To configure SSL, you must perform the following actions:

1. Generate security certificates for all distributed components (ACS, BOCS, DMS), WDK, and every WildFly server on which the ACS, BOCS, and DMS applications run.
2. Import those certificates into each of their keystores.
3. Enable SSL on all of their WildFly application servers and Content Server.

Note:

- These instructions use self-signed security certificates as examples. Self-signed certificates are adequate for testing or when all applications and users reside within a secure environment. However, in a production or a non-secure environment, you should use certificates that have been verified by a certification authority (CA). The CA might have instructions on generating and verifying security certificates, which might also involve using the Java `keytool.exe` utility.
- To create a self-signed security certificate and keystore for each ACS, BOCS, DMS, and WildFly installation, you use the Java `keytool.exe` utility. The *Oracle documentation* contains more information about `keytool.exe`. `keytool.exe` is installed with the JDK or JRE (for example, `C:\Documentum\java64`).
- The *Red Hat documentation* contains more instructions about configuring SSL on WildFly.
- To use AES256 and SHA256 algorithms, ensure that you have the JCE Unlimited Strength Jurisdiction policy files.

Generating and importing security certificates

1. Create a key and a local certificate keystore for all ACS, BOCS, DMS, and WDK Web applications as well as every WildFly server on which the ACS, BOCS, and DMS application run.

For example:

```
keytool -genkey -alias wildfly -keyalg RSA -keystore wildfly.keystore
```

You are prompted for additional information. These instructions use the following sample values:

Prompt/Parameter	Application	Sample Value	More Information
-alias	ACS	acskey	N/A
	BOCS	bocskey	
	DMS	dmskey	
	WDK	wdkkey	
	WildFly server	wildfly	

Prompt/Parameter	Application	Sample Value	More Information
-keystore	ACS	acs.keystore	N/A
	BOCS	bocs.keystore	
	DMS	dms.keystore	
	WDK	wdk.keystore	
	WildFly server	wildfly.keystore	
What is your first and last name?	ACS WildFly server	acsmachine .dnsname.com	Also known as the CN or common name. Specify the complete hostname of the application server as referenced by your browser or web service consumer.
	BOCS WildFly server	bocsmachine .dnsname.com	
	DMS WildFly server	dmsmachine .dnsname.com	
	WDK	wdkmachine .dnsname.com	
Enter keystore password	All	changeit	N/A

Note:

- The keystores and certificates can be created from any supported Java (Documentum bundled or external).
 - You should import the DMS certificate in `dfc.keystore` of Content Server. For example, `%DOCUMENTUM%\dba\secure`
- Copy `wildfly.keystore` to `%WildFly_HOME%\server\servername\conf`, where *servername* is the name of the ACS, BOCS, or DMS Web application.
 - Generate every Web application's certificate by exporting each one's keystore file. For example:

```
keytool -export -alias acskey -file acs.cer -keystore acs.keystore
```

These instructions use the following sample values:

Prompt/Parameter	Application	Sample Value	More Information
-file	ACS	acs.cer	N/A
	BOCS	bocs.cer	
	DMS	dms.cer	
	WDK	wdk.cer	

- Now import each Web application's certificate into all running Java programs on the hosts with which the Web application needs to connect as follows:

Application Certificate	Application Machine
ACS	ACS, BOCS

Application Certificate	Application Machine
BOCS	BOCS, DMS
DMS	BOCS, DMS, WDK

For BOCS pre-caching job, you should import the DMS certificate on Content Server. For example:

```
keytool -import -noprompt -trustcacerts -alias acscert -file
"C:\certificate\dms.cer" -keystore
"C:\Documentum\java64\jre\lib\security\cacerts"
```

Note:

- On the WDK client machine, import this certificate to the Java installation where the WDK application is running.

For example:

```
keytool -import -noprompt -trustcacerts -alias acscert -file
"C:\certificate\acs.cer" -keystore
"C:\Documentum\java64\jre\lib\security\cacerts"
```

- The default password for the cacerts Java trust store is the following:
changeit
 - On the ACS and BOCS machines, the default location of the Java programs installed by Documentum is \$Documentum\java64.
 - On a WDK client machine, the Java installation is located with the client application.
5. The ACS, BOCS and WDK client certificates—but not the DMS certificate—must also be imported into the UCF client's JRE and browser's Java program (where the WDK application URL is accessed) because if the UCF client's JRE version is higher than the existing JRE on the host, then the UCF client's JRE is installed during content transfer.

Note: The default location for UCF client's JRE is:

```
user-directory\Documentum\ucf\machine-name\shared\jreversion
```

- Use `keytool.exe` to import the ACS, BOCS, WDK certificates into the UCF client's JRE. For example:

```
keytool -import -noprompt -trustcacerts -alias acscert -file
"C:\certificate\acs.cer" -keystore
"C:\smithj\Documentum\ucf \smithj.acme.com
\shared\jre6\lib\security\cacerts"
```

- Use `keytool.exe` to import the ACS, BOCS, and WDK certificates into the browser's Java. For example:

```
keytool -import -noprompt -trustcacerts -alias acscert -file
"C:\certificate\acs.cer" -keystore
"C:\Program Files\Java\jre6\lib\security\cacerts"
```

Configuring web applications for SSL

This section describes how to configure web applications for SSL.

ACS

1. Modify the standalone.xml in:

```
%WildFly_HOME%\server\DctmServer_MethodServer\configuration\standalone.xml
```

For example, in standalone.xml, after

```
<subsystem xmlns="urn:wildfly:domain:web:1.1|urn:wildfly:domain:web:1.1"
default-virtual-server="default-host" native="false">
```

add the following:

```
<connector name="https" protocol="HTTP/1.1" scheme="https"
socket-binding="https" secure="true"
<ssl name="https" password="password" certificate-key-file=
"c:/jms.keystore" cipher-suite="TLS_RSA_WITH_AES_128_CBC_SHA"/> </connector>
```

Provide the keystore file path for "certificate-key-file" and password for "password" which you have set for keystore.

Note: JMS supports dual mode (anonymous and non-anonymous SSL) also when we have Content Server in anonymous SSL mode. So for dual mode, the cipher suite parameter should be

```
TLS_DH_anon_WITH_AES_128_CBC_SHA,SSL_DH_anon_WITH_3DES_EDE_CBC_SHA,TLS
_RSA_WITH_AES_128_CBC_SHA.
```

2. Restart ACS.
3. To validate the SSL configuration, navigate to the following URL:

```
https://acsmachine.dnsname.com:9082/ACS/servlet/ACS
```

"ACS Server is running" should be displayed.

BOCS

1. Modify the standalone.xml in:

```
%WildFly_HOME%\server\DctmServer_BOCS\configuration\standalone.xml
```

For example, in standalone.xml, after

```
<subsystem xmlns="urn:wildfly:domain:web:1.1|urn:wildfly:domain:web:1.1"
default-virtual-server="default-host" native="false">
```

add the following:

```
<connector name="https" protocol="HTTP/1.1" scheme="https"
socket-binding="https" secure="true"
<ssl name="https" password="password" certificate-key-file=
"c:/jms.keystore" cipher-suite="TLS_RSA_WITH_AES_128_CBC_SHA"/> </connector>
```

Provide the keystore file path for "certificate-key-file" and password for "password" which you have set for keystore.

2. Restart BOCS.
3. To validate the SSL configuration, navigate to the following URL:

```
https://bocsmachine.dnsname.com:8088/bocs/servlet/ACS
```

"ACS Server is running" should be displayed.

DMS

1. Modify the `standalone.xml` in:

`%WildFly_HOME%\server\DctmServer_DMS\configuration\standalone.xml`

For example, in `standalone.xml`, after

```
<subsystem xmlns="urn:wildfly:domain:web:1.1|urn:wildfly:domain:web:1.1"
default-virtual-server="default-host" native="false">
```

add the following:

```
<connector name="https" protocol="HTTP/1.1" scheme="https"
socket-binding="https" secure="true"
<ssl name="https" password="password" certificate-key-file=
"c:/jms.keystore" cipher-suite="TLS_RSA_WITH_AES_128_CBC_SHA"/> </connector>
```

Provide the keystore file path for "certificate-key-file" and password for "password" which you have set for keystore.

2. In `%WildFly_HOME%\server\DctmServer_DMS\deploy\DMS.ear\lib\configs.jar\dms.properties`, change the value of `dms.webservice.update.url` to `https://dmsmachine.dnsname.com:8491`
3. Restart DMS.
4. To validate the SSL configuration, navigate to the following URL:
`https://dmsmachine.dnsname.com:8491/dms-ws`

WDK Client Application Server

1. Modify `server.xml` on the application server to enable SSL and the server to use the `https` protocol. Check for SSL configuration lines in the `server.xml`, uncomment them and provide the appropriate details for the keystore file path, password, port, and so on.

For example, on Tomcat, uncomment the following lines in `app_server_home\conf\server.xml`:

```
<Connector
protocol="HTTP/1.1"
port="8443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="${user.home}/wdk.keystore" keystorePass="password"
clientAuth="false" sslProtocol="TLS"/>
```

2. Restart the application server.
3. To validate the SSL configuration, navigate to the following URL:
`https://wdkmachine.dnsname.com:8443/da`
where `da` represents the Documentum Administrator Web application.
The Documentum Administrator login page should be displayed.

Note: When performing an asynchronous write operation, DFC on the clients' (for example, Webtop) application server sends a message to DMS which is routed to the appropriate BOCS to inform how to upload the document. If DMS is configured in SSL mode, you should import the DMS certificate in the clients' application servers' keystore. For WebSphere, you should import the certificate in the default trust store.

Modify/configure ACS/BOCS/DMS configs

This section discusses modifying/configuring ACS/BOCS/DMS configs in the repository.

1. For the ACS server connection, specify the following attributes:

Attribute	Sample Value
Protocol	https
Base URL	https://acsmachine.dnsname .com:9082/ACS/servlet/ACS

These attributes are available in the ACS Server config.

2. For the BOCS server connection, specify the following attributes:

Attribute	Sample Value
Protocol	https
Base URL	https://bocsmachine.dnsname .com:8088/bocs/servlet/ACS

These attributes are available in the BOCS Server config.

3. For the DMS server's BOCS message routing, specify the following attributes:

Attribute	Sample Value
Post URL	https://dmsmachine.dnsname .com:8491
Consume URL	https://dmsmachine.dnsname .com:8491

These attributes are available in the DMS Server config.

Troubleshooting the SSL Configuration

1. Check the ACS, BOCS and DMS server logs to make sure that no SSL-related errors or exceptions have occurred.
2. Verify that you can access the ACS, BOCS, DMS, and WDK client application URLs from every other host by using their full hostnames with the HTTPS protocol and SSL port. These URLs must be accessible from every host.
3. Use `keytool.exe` to verify that each certificate has been imported into the trusted store. For example:

```
keytool -list -keystore "jre_path\lib\security\cacerts"
```


Installing remote Content Servers in distributed or load-balanced configurations

This section provides instructions for installing and configuring remote Content Servers in distributed or load-balanced content configurations.

If you are creating a new, single repository in a distributed or load-balanced content configuration, a configuration program separate from the Content Server configuration program is used for installing remote Content Servers and creating the storage areas on the remote hosts and related location objects.

Preinstallation requirements

The following requirements and limitations exist when you are installing remote Content Server in distributed or load-balanced configurations:

- The remote host must meet the same preinstallation requirements as the primary Content Server host.
- The remote Content Server must be installed in the same installation directory as the primary Content Server.
- The database client software must be installed on remote Content Server hosts. The remote Content Server configuration program must connect to the database to properly create the following objects for the remote server.
 - server config
 - acs config
 - file store storage
 - location
- When a remote Content Server is created for a distributed or load-balanced content environment, the server.ini file from the primary Content Server host is copied from the primary host to the remote host. The values used on the primary and remote hosts for database connectivity must be identical to ensure that the value of the database_conn key on the primary Content Server host is valid on the remote hosts. For example, if the database is SQL Server, ensure that the DSN name for the SQL Server instance's ODBC data source is the same on all hosts.
- All hosts in a distributed or load-balanced configuration must be set to the same UTC time.
- A repository that uses a distributed or load-balanced storage area with encrypted file stores as components cannot use shared content.



Caution: After a repository has been configured to use distributed or load-balanced storage, it is not possible to revert to using non-distributed storage.

- The following requirements must be met when the Content Server file store is assigned to a shared folder on the network with a UNC path:
 - Content Server and the file store must be on the same domain.
 - The installation user account of Content Server must be available on the domain.
 - The installation user account must have full access control for the file store.

Installing and configuring the remote Content Server

To configure the remote Content Server:

1. Copy the Content Server installation files from the installation media to the correct directories on the host.

This step is identical to the process used to copy files onto the primary Content Server host. *EMC Documentum Platform and Platform Extensions Installation Guide* contains more information.

2. Install the Content Server software. At the end of the installation, do not choose the **Configure Now** option and exit the installer.

EMC Documentum Platform and Platform Extensions Installation Guide contains more information.

Note: When installing remote Content Server, if the remote connection broker is configured for Certificate-based mode then you need to manually copy all the certificate files including lockbox file to the remote Content Server machine.

3. Run one of the following files:

- Windows: %DM_HOME%\install\cfsConfigurationProgram.exe

The configuration program starts automatically following a reboot of the host. However, if it does not start automatically or if you have to delete the remote Content Server and must reconfigure the remote Content Server, simply rerun it.

- UNIX and Linux: \$DM_HOME/install/dm_launch_cfs_server_config_program.sh

4. (Windows only) Type the installation owner's password.

5. Specify the hostname of the primary connection broker for the repository and verify or type the port number on which the connection broker listens.

The port defaults to 1489. If you are using the default port number, ensure that the next port number (1490) is available for use because two ports must be reserved for the connection broker.

Select **Use certificates** if primary Content Server and repository are configured with SSL certificates.

Provide the DFC trust store information:

- **TrustStore:** The location of the DFC trust store. Before that you need to copy the DFC trust store from primary Content Server to remote Content Server location.
- **Password:** The password of the trust store file.

Select **Use Default Java TrustStore** if you want to use the default DFC Java trust store.

Click **Next**.

6. Specify the fully qualified domain name (FQDN) of the remote Content Server host.
7. Select the repository for which you are installing the remote Content Server and enter the primary Content Server installation owner and domain information.
The repository list is populated with repositories known to the connection broker for which you provided information in the previous step.
8. Enter the name and port number for the connection broker on the current host, indicate whether connection broker startup following a system restart is automatic or manual.
The default values are `Docbroker` and `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. The connection broker is started.
9. Accept the default location of the data directory or browse to a different location.
The data directory is where content files are stored in the repository.
10. Accept the default location of the share directory or browse to a different location.
The share directory is where clients, example code, and required libraries are stored.
11. Accept the default service name for the new remote Content Server or type a different name.
12. Enter the global repository login name and password.
13. Installation is complete.

Note: If the primary server uses an LDAP Server to authenticate, copy the file `Documentum\dba\config\ldap_XXXXXXXXXXXXXXXXX.cnt` from the primary to the remote server (XXXXXXXXXXXXXXXXX represents the `r_object_ID` of the LDAP config object).

To start the application server instance that is running the Java Method Server and Content Server, perform one of the following actions:

- On Windows, restart after the installation.
 - On distributed or load-balanced UNIX and Linux configurations, use Documentum Administrator to set the `Get` method for each component of the distributed or load-balanced store to `Surrogate Get`.
14. If required, modify the `dm_server_config` object to specify only the `app_server_name` and `app_server_uri` entries that are relevant to the remote Content Server.

Note: Because the remote Content Server installation clones a copy of the `dm_server_config` object from the original repository, unnecessary attribute values might have been copied over. For example, if an index agent and Business Process Engine are in the original repository, you might have entries for them that point to the original host machine on the remote host. Remove any of these attributes if they are not applicable to the new remote Content Server installation.

Upgrading a distributed or load-balanced configuration

EMC Documentum System Upgrade and Migration Guide contains the detailed information.

Deleting a remote Content Server

Use these instructions to delete a remote Content Server and its software installation in a distributed or load-balanced content environment. These instructions delete only the remote Content Server. They do not delete the repository or affect the primary Content Server for the repository.

Note: On Windows, do not use the Server Manager program to uninstall the server because the Server Manager program launches the configuration program for primary Content Servers (not remote Content Servers).

Before deleting the software installation, delete any connection brokers on the host.

To delete a remote Content Server and its server software installation:

1. Log in to the host as the Content Server installation owner.
2. Run the remote Content Server configuration program in one of the following locations:
 - Windows: %DM_HOME%\install\cfsConfigurationProgram.exe
 - UNIX and Linux: \$DM_HOME/install/dm_launch_cfs_server_config_program.sh
3. (Windows only) Enter the installation owner's password.
4. Select **Delete remote Content Server**.
5. Enter the installation owner's name and password.
6. To delete its server software installation, run the server uninstaller in one of the following locations:
 - Windows: %Documentum%_uninst\Server\uninstall.exe
 - UNIX and Linux: \$DOCUMENTUM/uninstall/server/uninstall/uninstall.bin

Implementing single-repository models

This section describes how to implement the building blocks most commonly used to create single-repository distributed configurations.

Implementing a distributed repository without a distributed storage area

Use the instructions in this section to implement the single-repository distributed model. This model is the preferred model for single-repository distributed configurations if users are accessing the repository using web-based clients.

To set up a distributed repository without a distributed storage area:

1. If not already installed, install the primary site by following the instructions in the *EMC Documentum Platform and Platform Extensions Installation Guide*.

Note: One repository in the installation must be designated as the global registry if you installing BOCS at the remote sites.

2. Install DMS on a separate host at the primary site if you are installing BOCS at the remote sites. Use the instructions in the *EMC Documentum Platform and Platform Extensions Installation Guide* to install and configure the server.

3. To use a BOCS server at each remote site, install BOCS.

4. Define the network locations for the remote sites.

If the remote sites are accessing content through the ACS server at the primary site and you are not installing BOCS servers at the remote sites, defining network locations is not required.

Use Documentum Administrator to define network locations. You must be connected to the repository designated as the global registry to define network locations.

Installing with distributed storage areas

The information in this section is useful when setting up distributed configurations that use a distributed storage area and no Documentum installation is present. This section does not provide instructions for converting an existing installation. Following this procedure results in a single repository that uses a distributed storage area distributed across multiple geographical locations.



Caution: After a repository is set up with a distributed storage area and is using it, you cannot remove it and return to a standalone, non-distributed configuration.

There are three stages in the process of setting up a distributed storage area:

1. Planning
2. Set up the primary site (the site where the RDBMS resides)
3. Set up the remote sites

Planning

Before you begin installing the Content Servers at any site:

- Decide whether to share distributed content, replicate distributed content, or use a combination of both.
- Read and follow the guidelines in the next section to ensure that your environment is set up correctly.
- Use our guidelines for estimating the required disk space to make sure that you have enough disk space at the site.

Guidelines

To ensure that your distributed architecture works properly, follow these guidelines:

- The Content Server at each site (primary and remote) must be able to authenticate the user, using the same mechanism.

When a remote user logs on to a repository, the client sends two connection requests, one to the remote content server and one to the data server. Each server must be able to authenticate the user using the same authentication mechanism.

- If you intend to share content files among the component storage areas, the installation owner for all servers accessing the repository must be the same account at all sites.

Having the same installation owner at each site allows the Content Server at each site to access content files at the other sites. On Windows platforms, to meet this requirement, have a global domain for all sites and establish a global `dmadmin` account (or equivalent) in that domain. At each site, log on to the global `dmadmin` account when you install and configure the Content Server.

- On Windows, all machines running a Content Server for a particular repository have to be in the same domain.
- If the sites are not connected using NFS, method objects must be resolvable at all server sites .

In a distributed installation, the method commands defined by the `method_verb` property of the method object must exist at each server site.

Note: Some method objects might not have a full file system path defined (in the `method_verb` property) for the program they represent. For such programs to work correctly, the command executable must be found in the PATH definition for the user who is executing the command. If the `run_as_server` property for the method object is set to `TRUE`, the user executing the command is the installation owner. If `run_as_server` is set to `FALSE`, the user executing the command is the user who has issued the `EXECUTE` statement or the `DO_METHOD` administration method. (`run_as_server` is set to `FALSE` by default in the methods defined in the `headstart.ebs` file.)

Estimating disk space

Before you begin installing distributed sites, estimate how much disk space is required for content storage at each site.

The amount of space required at each distributed site depends on the following factors:

- Total number of bytes per document
- Total number of documents in the distributed repository
- Number of distributed sites
- Number of versions of each document that you intend to keep online
- Whether you intend to keep renditions of the documents
- Whether you intend to replicate each document to all sites

The formula for estimating disk space is:

```
(total # of documents) x (# of sites)x(bytes/document) x (# of versions)
= total amount of disk space
```

Estimating document size

To estimate the total number of bytes per document, sum the following figures:

- Number of bytes for an average document
- Number of bytes for any renditions

An example of disk space calculations

To illustrate estimating disk space, assume the following:

- Your enterprise has three distributed sites
- Site 1 has 10,000 documents
- Sites 2 and 3 has 5,000 each
- You intend to index PDFText renditions of the documents
- Each document is an average of 10K bytes.

First, estimate the total number of bytes per document. Include in the total the estimated size of the document's content and renditions. For example, assume that each document has 10K of content and PDF and PDFText renditions. For a 10K document, the PDF rendition is approximately 8K and the PDFText rendition is approximately 6K. Sum these estimates to arrive at the estimated total number of bytes per document. In our example, the sum is 24K. Use this total in the disk space formula to determine the disk space needed at each site.

In this example, the three sites have a total of 20,000 documents at 24K per document. Three versions of each are kept online, with each replicated at each site:

20,000 docs x three sites x 24K/doc x three versions is approximately 4.68 gigabytes

This calculation indicates that you need a total of 4.68 gigabytes of disk space at each distributed site.

Setting up the sites

The procedure in this section describes how to install a distributed storage area as part of a new Documentum installation. It does not provide instructions for adding a distributed storage area to an existing installation.

Installing a remote site installs a remote content server and an ACS server and creates a local file store storage area for the site. The name of the local file store has the following format:

`fs_rcs_server_config_name`

where `server_config_name` is the name of the server configuration object for the site's remote content server. If the full name, including the `fs_rcs_` prefix is longer than 32 characters, the prefix is truncated to 32 or fewer characters.

The installation procedure also runs a script to install all the administration methods needed for that storage area.

To implement distributed storage:

1. Make sure that you have read and are complying with the guidelines in the previous section.
2. Decide how much disk space you need at each distributed site to store content files.
Read the instructions on estimating disk space requirements.
3. At the primary site (where the RDBMS is located), install Content Server and configure the repository.
EMC Documentum Platform and Platform Extensions Installation Guide contains more information.
4. Install the index agent and index server at the primary site also.
Use the instructions in the *EMC Documentum xPlore Installation Guide*.
5. Install DMS on a separate host at the primary site if you are installing BOCS at the remote sites.
6. Start Documentum Administrator and connect to the repository as a user with superuser privileges.
7. Create the component storage area for the primary site.
Use Documentum Administrator to create the storage area.
To ensure backwards and future compatibility, do not use the default file store, `filestore_01`, as the distributed component at the primary site. Create a storage area to be the distributed component.
It is recommended that you place the content store on a different drive from the Documentum installation. This action allows you to separate Documentum programs from Documentum data.
8. Create the distributed storage area.
Use Documentum Administrator to create the distributed storage area.
Be sure to add the storage area you created in the previous step as a component of the distributed storage area. (Do not use `filestore_01` as the distributed component.)
If you intend to implement content sharing for any or all of the content files in the distributed storage area, do not check **Fetch Content Locally Only**.
If any of the following conditions exist, check **Fetch Content Locally Only**:
 - If all the files are replicated among the distributed storage area components
 - If the components are encrypted file stores
 - If you to use Surrogate GetChecking **Fetch Content Locally Only** sets `only_fetch_close` to TRUE, which restricts the server's read access to its local component storage area. It is not able to fetch from the other component areas.
9. Set the default storage area for SysObjects to the distributed storage area.
Use type management facilities in Documentum Administrator to set the default storage area for SysObjects.
10. Move the objects currently in `filestore_01` to the distributed store.
Note: Objects in `filestore_01` are created by default during Content Server installation. These objects must be in the distributed store's component storage areas for EMC Documentum desktop to function properly. However, you cannot make `filestore_01` a component of the distributed store to resolve this issue. Doing so does not reset the properties in the objects correctly. Additionally, it is recommended that you do not make `filestore_01` a component.

Use a DQL `UPDATE...OBJECTS` statement to move the objects:

```
UPDATE dm_sysobject (ALL) OBJECTS SET a_storage_type = 'name_of_distributed_store' WHERE a_storage_type = 'filestore_01'
```

`name_of_distributed_store` is the name of the distributed storage area you created.

Note: If the file store storage area has been in use already, it might have objects in it, such as installed lifecycles, that cannot be moved. The previous query fails in such circumstances. This situation can occur if you are adding a distributed storage area to an existing repository that has been in use for any time. To resolve this issue, either modify the query to exclude such objects or, in the case of lifecycles (dm_policy objects), make sure that they are removed first.

11. Set the timeout value for the server at the primary site to a minimum of 30 minutes.

The primary server's connection with a remote desktop client can time out while content is being transferred between the client and the remote content server. If a timeout occurs during a save operation on a new document, the primary server does not save the new object to the repository. To prevent this occurrence, set the `client_session_timeout` key in the [SERVER_STARTUP] section of the `server.ini` file to a minimum of 30 minutes. The Documentum Administrator online help contains the information on modifying the `server.ini` file.

12. At each remote site, use the instructions in *EMC Documentum Platform and Platform Extensions Installation Guide* to install a remote content server.

Note: All servers must be trusted servers or all servers must be non-trusted servers. You cannot mix trusted servers and non-trusted servers against one repository.

13. Log on to Documentum Administrator to perform the following manual steps:

- a. Update the distributed storage area to include the file store storage areas created at the remote sites.

- b. Update the **Far Stores** list in each site's server configuration object to include the component storage areas for the other sites.

For example, suppose there are three component sites, SiteX, SiteY, and SiteZ. If you are editing the server configuration object for siteX, its **Far Stores** list must include the component storage areas for SiteY and SiteZ.

- c. (Optional) Reset the proximity values for the remote content servers.

The installation process automatically sets a remote server's proximity to its local connection broker as 9001 and its proximity to the primary Content Server as 9010. You might want to reset the values to reflect your network's topography.

14. Ensure that servers at all sites can authenticate users and groups accessing distributed documents, using the same mechanism.

In a configuration using a distributed storage area, when a user connects to a repository, the client sends two connection requests: one to the data server and one to the remote content server. Each server must be able to authenticate the user's name and password using the same authentication mechanism.

15. If users at remote sites are accessing the repository using web-based client applications, perform the following additional steps for each remote site:

- a. Define at least one network location for the ACS server at the site.
- b. Add the network location to the ACS configuration object.

Add the location to the ACS configuration object for each ACS server that can service requests from that network location.

- c. Define the network location's proximity to the ACS server or servers.
 - d. Define the ACS server's projection to a connection broker.
16. Ensure that all machines hosting a Content Server, remote content server, or ACS server are using UTC time and are synchronized.

When a server at a remote site (remote content server or ACS) connects to the primary site to fetch content using surrogate get, it uses a global login ticket configured to be valid for 5 minutes to connect. If the clocks in the host machines are not synchronized, the connection can fail if the host machine finds the login ticket invalid if the time difference is greater than 5 minutes.

The dm_rcs_setup.ebs script

The script `m_rcs_setup.ebs` is executed automatically when you install a remote site. It creates the administration jobs necessary to manage and administer the remote site's file store storage area, such as Content Replication, `dmclean`, and `dmfilescan`. It also enables Surrogate get for the storage area at the remote site and creates a `sysadmin` log directory on the remote host.

You can run this script manually. (Rerunning the script in an installation in which it has already been executed is harmless.) The command line is:

```
dmbasic -f %DM_HOME%\install\admin\dm_rcs_setup.ebs -eRCSSetup --  
repository server_config_name local_connection_broker_host local  
_connection_broker_port
```

where:

- *repository* is the name of the repository.
- *server_config_name* is the name of the server configuration object for the Content Server at the remote site.
- *local_connection_broker_host* is the host name of the computer for the local connection broker.
- *local_connection_broker_port* is the port where the local connection broker listens.

Creating network locations

Use the instructions and information in this section to create network locations after you install the servers supporting a web-based, single-repository, distributed configuration.

Use Documentum Administrator to create network locations. The Documentum Administrator online help contains the instructions.

Note: Do not create or modify a network location using DQL or the API. The ACS server does not recognize changes made through DQL or the API.

Network locations represent places or portions of a network's topography. Typically, each network location is defined as one or more IP addresses or range of addresses that are in that location. The addresses identified in network locations can overlap. That is, one or more network locations can include the same IP address or address range. This situation provides an opportunity for users to

choose their network location when they start sessions with web-based clients. The clients can be configured to present the users with a list of all network locations configured for their IP address.

However, defining an IP address or address range in a network location is optional. If a client application fails to find a network location that includes a user's IP address, the application assigns the user to a default location. The application can also present the user with a list of default locations and allows the user to choose. If there are no defaults, the ACS at the primary site services content requests. A Boolean property in the objects that record network location definitions identify default network locations.

By default, each ACS server uses all of the network locations that are configured for the associated Content Server. For example, suppose a remote office in Florida has a Content Server and an ACS server. All IP addresses for machines in that office and the ones of the Florida-based telecommuters would be specified in one network location map object. That object is then identified in the ACS configuration object associated with the ACS server in the Florida office. An ACS (and BOCS) server can also serve multiple network locations. For example, perhaps a separate network location object is defined for each building in the Florida office and one for each telecommuter.

Network locations are recorded in the installation as `dm_network_location_map` objects that reside in the global registry. They must be stored in the global registry. The name of each location, stored in the `dm_network_location_map.netloc_ident` property, must be unique among the set of network locations in the global registry. If network locations are manually added to the ACS configuration object, they are identified in the `acs_network_locations` repeating property. Within a BOCS configuration object, they are identified in the `network_locations` property. If you want the locations to be available to the BOCS server, add them to the BOCS configuration objects.

Adding network locations to an ACS or BOCS configuration object

To add a network location to an ACS or BOCS configuration object, perform the following actions:

- Identify the network location as a location serviced by the ACS or BOCS server represented by the configuration object.
- For ACS only, specify the network location's proximity to the server.

An ACS server can service multiple network locations, but it can be closer to some of those locations than it is to others. Set the network location's proximity value for each ACS server appropriately.

Use Documentum Administrator to update an ACS configuration or BOCS configuration object to add a network location and its proximity values.

Projecting an ACS server to connection brokers

ACS servers must project their presence to at least one connection broker. Content Server uses that information to determine which ACS servers are running. If the ACS configuration object represents a BOCS server, it is not necessary to set these properties.

The following table lists the properties used to project presence to a connection broker. These properties are repeating properties and the values in one index position across the properties represent the settings for the projection to one connection broker.

Table 10. Properties in ACS configuration objects related to connection broker projection

Property	Use
projection_enable	<p>Determines whether the ACS server is projecting to the connection broker identified in the corresponding index position in <code>projection_target</code>.</p> <p>Set to T (TRUE) to project to the connection broker.</p> <p>Note: If this property is set to true for a particular index position, then <code>projection_netloc_enable</code> at the same index position must be false.</p>
projection_targets	<p>List of the connection brokers to which the ACS server projects its presence and proximity.</p> <p>Set this property to the name of the connection broker.</p>
projection_ports	<p>List of the ports on which connection brokers are listening. The port specified at a particular index position corresponds to the connection broker identified at the corresponding index position in <code>projection_targets</code>.</p>

Setting ACS proximity values for network locations

Proximity values for network locations represent the location's proximity to an ACS server. By default, the ACS server installed at the primary site has a built-in proximity of 9001. Therefore, by default, all network locations have a proximity of 9001 to the ACS server at the primary site. If a user at a network location cannot access other, closer ACS servers, the ACS server at the primary site is used.

Proximity values are not set in the network location object, but instead are set in the ACS configuration object. Alternatively, the proximity values defined in the server configuration object for the Content Server associated with the ACS server can be used.

Use Documentum Administrator to edit the ACS configuration or server configuration object. The Documentum Administrator online help contains more information.

Table 11. Properties in ACS configuration and server configuration objects related to network proximity values

Property	Description
projection_netloc_enable (dm_server_config and dm_acs_config)	<p>Determines whether the system regards the ACS server as available for the network location identified in the corresponding index position in projection_netloc_ident.</p> <p>Set this property to T (TRUE) to make this ACS server accessible to users in the network location identified in the corresponding index position in projection_netloc_ident.</p> <p>If this property is F (FALSE), the system assumes that the ACS server is currently unable to handle requests originating in the corresponding network location.</p> <p>Note: If this property is true for a particular index position, then projection_enable at the same index position must be false.</p>
projection_netloc_ident (dm_acs_config) projection_netloc_id (dm_server_config)	<p>List of the network locations that the ACS server can service. Set this property to the name of the network location. This property is the name set in the netloc_ident property of the dm_network_location_map object.</p> <p>Note: Network locations specified in this property must also be listed in the dm_acs_config.acs_network_locations property.</p>
projection_proxval (dm_server_config and dm_acs_config)	<p>Proximity of the network location at the corresponding index position in the projection_netloc_ident property to the ACS server.</p> <p>In an ACS configuration object, set this property to a number from 1 through 8999. The lower the number, the closer to the ACS server the network location is presumed to be.</p> <p>In a server configuration object, the number might be over 9000. The system adjusts the value to use only the actual proximity, the value minus the 9000. (The number might also be being used to define a remote server's proximity.)</p>

Defining accessible storage areas for an ACS server

Which storage areas an ACS server can access is dependent on the value in its config_type property:

- If config_type is set to 1, the ACS server uses the storage areas listed in the server configuration object's far_store property as a list of inaccessible storage areas. That is, if a storage area is identified in the far_stores property, the ACS server cannot fetch content from that storage area. Instead, the ACS server can fetch content from any storage area visible to the server and not listed in that property.

Note: When the config_type is 1, the ACS Properties page in Documentum Administrator displays those storage areas listed in the Content Server's far_stores property. You cannot edit

that field from the ACS Properties page. Edit it from the server configuration's Properties page. If the `config_type` is 1, **Use the projection targets, network locations, and near stores already configured for the associated server configuration object** is selected on the ACS Properties page.

- If the `config_type` property is set to 2, the ACS server uses the storage areas listed in the `near_stores` property of its ACS configuration object as its accessible storage areas. If the `config_type` is 2, **Manually enter projection targets, network locations, and near stores** is selected ACS Properties page.

The `near_stores` property is not set during the installation process.

By default, `config_type` is set to 1 when an ACS server is installed. To change the setting, and to set the `near_stores` property, use Documentum Administrator to edit these values from the ACS server Properties page.

Accessing file stores in a distributed environment

Set the `is_public` parameter to True and mount the file system in a distributed environment to gain access to the file stores. Otherwise, an error message appears indicating that the current Content Server cannot access the storage area.

Modifying an `acs.properties` file

The `acs.properties` file contains configuration information for the ACS or BOCS server with which it is associated. The `repository.acsconfig` entry in the file identifies the server associated with the file. Use a text editor to modify the `acs.properties` file.

When the associated server is an ACS server, the location is as follows:

- Windows:
`%DOCUMENTUM%\wildfly_directory\domains\DctmDomain\upload\MethodServer\acs.ear\APP-INF\classes\config`
- UNIX and Linux:
`$DOCUMENTUM_SHARED/wildfly_directory/domains/DctmDomain/upload/MethodServer/acs.ear/APP-INF/classes/config`

When the associated server is a BOCS server, the location is as follows:

- Windows:
`%DOCUMENTUM%\wildfly_directory\domains\DctmDomain\upload\BOCS\bocs.ear\APP-INF\classes\config`
- UNIX and Linux:
`$DOCUMENTUM_SHARED/wildfly_directory/domains/DctmDomain/upload/BOCS/bocs.ear/APP-INF/classes/config`

The mode.cachestoreonly entry

This key indicates whether the acs.properties file is associated with an ACS or BOCS server. The value in this key must match the value specified in the is_cache_acs property of the ACS configuration object for the server. The values of both these items are set when the ACS or BOCS server is installed.

Do not change these values. If the values of the key and the ACS configuration property do not match, the server does not work. If the values are incorrect for the server, the server does not behave appropriately.

Adding entries for additional servers

If you are updating the file to allow an ACS server to communicate with an additional server for a repository in an installation, set the following configuration parameters for the server:

```
repository.name
# name of the repository
repository.login
# login name to be used to connect the server
repository.acsconfig
# name of the acs config object created for the server
```

Each parameter entry of a particular type is numbered after the first. For example, suppose the file currently has entries for only one server, the server for the Engineering repository. You want to add entries for an additional server for that repository. The additional entries would be:

```
repository.name.1=enr_1 repository.login.1=enr_1_login_user
repository.acsconfig.1=<acs config object name>
```

Each additional entry for each parameter is designated with a 1. If you added a third server, the number increments by one. The entries for the third server would end in 2 (The first entry for each parameter has no number.) You can add up to 99 entries for these parameters.

You must specify the password in repository.password.# that corresponds to the user in repository.login.#. The password must be encrypted. Use the DFC password encryption utility to encrypt the password; for example:

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

Disabling access to an ACS server

You can disable access to an ACS server by particular network locations or by all network locations using that server. Use Documentum Administrator to disable access to an ACS server by network location.

To disable a particular network location's access to a particular ACS server:

- If the config_type property in the ACS configuration object is set to 2, set the projection_netloc_enable property in the ACS configuration object to F at the index position

associated with that network location. For example, if you set `projection_netloc_enable[2]=F`, then access from network location identified in `projection_netloc_ident[2]` is disabled.

- If the `config_type` property in the ACS configuration object is set to 1, set the `projection_netloc_enable` property in the associated server configuration object to F at the index position associated with that network location. For example, if you set `projection_netloc_enable[2]=F`, then access from network location identified in `projection_netloc_id[2]` is disabled.

To disable all access to the server, from any network location, set the ACS configuration object's `config_type` property to 0.

Configuring shared content files

To enable content file sharing among a distributed storage area's components, you must:

- Configure each component storage area directory as a shared directory before you add it to the distributed storage area.
- Set the `far_stores` property for the server configuration object at each site.
- Set the `only_fetch_close` property of the distributed storage area to FALSE.

The `far_stores` property in a server configuration object defines the distributed storage area components that are not local for that server. For example, suppose a distributed site has three locations: London, New York, and Paris. At the New York site, the `far_stores` property for the server is set to London and Paris. In London, the property is set to New York and Paris. And finally, in Paris it is set to New York and London.

Note: If you followed the installation procedure for distributed storage areas, the `far_stores` property for the server at each of your distributed sites contains the names of all the other sites.

A server cannot save to a storage area defined as far. The `only_fetch_close` property controls whether the server can fetch from a far storage area.

If the storage areas are shared directories and the property is FALSE (the default), the server can fetch files from far storage areas directly.

When `only_fetch_close` is TRUE, servers cannot fetch content files from component storage areas named in the `far_stores` property in their server configuration objects. If a user requests a document that is only in a far storage area, the server satisfies the request using surrogate get.

Creating pre-caching content jobs

A pre-caching content job generates a pre-caching request that is sent to the DMS server. The arguments in the job specify a query to identify the content you want to pre-cache and the network locations for which you want to cache the content. When the job executes, it executes the query and post a pre-caching request to the DMS. The job does not transfer any content for caching itself. It only posts the request for pre-caching to the DMS server. The DMS server passes along the request to the appropriate BOCS servers, which in turn, initiates the pre-caching operation.

Use Documentum Administrator to create a pre-caching job. A pre-caching job executes the `dm_PreCacheContent` method. The following table lists the arguments supported by the method.

Table 12. dm_PreCacheContent method arguments

Argument	Description
<code>docbase_name</code>	Name of the repository that contains the content you wish to pre-cache
<code>user_name</code>	Installation owner user name
<code>query_type</code>	Name of the object type that is the target of the DQL query
<code>query_predicate</code>	DQL WHERE clause qualification that identifies the content to be pre-cached
<code>network_locations</code>	Comma-separated list of network location object IDs or the keyword <code>dm_all_network_locations</code> . When the keyword is specified, the method retrieves all available network location object IDs.
<code>job_id</code>	Object ID of the job invoking the method
<code>cutoff_time</code>	(Optional) Defines a cutoff time for content. Only content that satisfies the predicate and created after this time is cached. The value specified in <code>cutoff_time</code> is compared with the value in the <code>set_time</code> property of the <code>dmr_content</code> object. The format of the value in this argument is: <code>mm/dd/yyyy HH:MM:SS</code>
<code>expiration</code>	(Optional) Length of time, in seconds, for which the messages generated by this job are valid. The default value is 3600 seconds (1 hour).
<code>batch_size</code>	(Optional) Number of objects processed in each request. For example, if the predicate returns 150 object, and the batch size is 50, the method processes the results in three batches, 50 objects in each batch. The default value is 50.

When the method executes, it finds all content files that satisfy the specified predicate and have a `set_time` value that is equal to or greater than the `cutoff_time` argument. The underlying query orders the results in ascending order, based on the `set_time` value. For the resulting content files, it sends messages to DMS requesting pre-caching for the content.

Each time the job completes, it resets the `cutoff_time` argument to the `set_time` value of the last content object processed by the method. In this way, the job does not send multiple pre-caching requests for the same content.

If the job stops before processing all returned content files, it resets the `cutoff_time` argument to the `set_time` value of the last processed content file. When the job is restarted, it resumes with the next content file.

If a content fails to pre-cache after the DMS delivers the pre-caching request to a BOCS server, then you can run the method manually with a custom predicate to regenerate the request for that content. Use Documentum Administrator to run the method.

Setting up content replication

Replicating content files among distributed storage area components ensures that users at each site have local copies of the files to access. Content Server includes several tools for content replication. You can set up replication to run automatically or you can perform the operation manually.

Deciding which tool to use

Deciding which tool to use based on how often replicated content changes and how important it is for users at any site to have access to current versions of the files.

Automatic Replication

There are two ways to replicate content automatically. You can use either or both in a single-repository distributed configuration. The tools are:

- ContentReplication tool
- Surrogate get feature

The ContentReplication tool greatly simplifies administration of content replication. The tool requires only enough temporary disk space to transfer the largest individual content file to be replicated. (In contrast, using dump and load requires enough temporary space to hold a dump file containing all content files to be replicated.) The ContentReplication tool is recommended for most situations. The Documentum Administrator online help contains the instructions on using this tool.

The surrogate get feature replicates content on demand to the user's local storage area. When a user attempts to access the content, the server recognizes that the content is not available locally and calls the `dm_SurrogateGet` method. The method obtains the content from another server and executes an `IMPORT_REPLICA` call on the local server to make the content available to the user. The `dm_SurrogateGet` method is a system-defined method installed with Content Server.

Manual replication

To replicate content files manually, you can use:

- The `REPLICATE` administration method
- The `IMPORT_REPLICA` administration method

The `REPLICATE` administration method copies a file from one storage area to another. The disks on which both component storage areas reside must be accessible to the server.

The `IMPORT_REPLICA` administration method imports a file from an external file system into a storage area. You can use `IMPORT_REPLICA` with or without NFS access.

You can execute either `REPLICATE` or `IMPORT_REPLICA` from Documentum Administrator or using the `EXECUTE` statement or the Apply method. On UNIX and Linux platforms, both source and target storage areas must be available to the server through NFS.

Using the surrogate get feature

When a user tries to access a content file that is stored in a remote component of a distributed storage area, the server must either fetch the file from the remote area directly or invoke the `dm_SurrogateGet` method to fetch the file and replicate it to the local storage area.

Use surrogate get if your site is not using shared directories. Read the instructions if you are using shared directories.

To use surrogate get, you must:

- Set the `far_stores` property for the server configuration object at each site.
- Set the `only_fetch_close` property of the distributed storage area to `TRUE`.
- Make sure the `get_method` properties in the storage areas are set to the name of the surrogate get method.
- Set the clocks on participating host machines to UTC time and synchronize the clocks.

Surrogate get uses a login ticket to connect to servers to obtain content files. The login tickets are valid for 5 minutes. If the machine on which the `dm_SurrogateGet` method is running and the host machine of the server to which it is connecting are not synchronized in time, the ticket can be invalid on the target. The ticket becomes invalid because of perceived differences in time. In such cases, the job fails.

The `far_stores` property in a server configuration object defines the distributed storage area components that are not local for that server. For example, suppose a distributed site has three locations: London, New York, and Paris. At the New York site, the `far_stores` property for the server is set to London and Paris. In London, the property is set to New York and Paris. And finally, in Paris it is set to New York and London.

Note: If you followed the installation procedure for distributed storage areas, the `far_stores` property for the server at each of your distributed sites contains the names of all the other sites.

The `only_fetch_close` property controls whether a server can fetch content from a component storage area named in the `far_stores` property. When `only_fetch_close` is `TRUE` (the default), servers cannot fetch content files from component storage areas named in the `far_stores` property in their server configuration objects. If a user opens a document whose content file is only in a far storage area, the server invokes the `dm_SurrogateGet` method to retrieve the file. This method is identified in the storage area's `get_method` property.

The `dm_SurrogateGet` method

The `dm_SurrogateGet` method implements the surrogate get feature. The method is installed automatically along with the system administration tools. When you create a storage area, the `get_method` property of the storage area object is set to `dm_SurrogateGet`.

By default the `dm_SurrogatGet` method requires the following arguments to control whether to perform a content check:

- `do_content_check`, which verifies the hash values of the content files and performs a content check.
- `no_content_check`, which does not perform a content check.

The method uses a global login ticket to connect to remote servers. The ticket has a validity period of 5 minutes. Host machines must be set to UTC time and synchronized. If that is not done, the host machine of the process receiving the login ticket can consider the ticket invalid if the time difference is greater than 5 minutes.

All surrogate get operations can be traced. When tracing is turned on, the trace file is called `sugg.trc` and is found in the `%DOCUMENTUM%\bin` (Windows) or `$DOCUMENTUM/bin` (UNIX and Linux) directory.

You can also turn on tracing for the method invocation. Tracing the method invocation logs the method command line in the server log whenever the method is invoked.

Using REPLICATE

The `REPLICATE` administration method copies a file from one component storage area to another. The arguments determine which file or files are copied and where the copies are put. *EMC Documentum Content Server DQL Reference Guide* contains the syntax.

You cannot select a component storage area from which to copy the file or files. Select only the target area. The server searches the remaining component areas in the distributed storage area and replicates the first copy of the file it finds.

Use the `EXECUTE DQL` command to execute this administration method as follows

```
EXECUTE replicate WITH query = 'dql_predicate', store = 'target_storage_area_name'
```

Using IMPORT_REPLICA

The `IMPORT_REPLICA` administration method copies a file into a distributed storage component storage area. The file can be located in another component of the distributed storage area or in an external file system. If the source file is in another component of the distributed storage area, the server must have access to the disk on which the area resides. On UNIX and Linux, if the disk is on another machine, the server must have access to NFS. If the file is in a directory on the same machine as the server or on a tape, diskette, or floppy, NFS is not required.

Use the `EXECUTE DQL` command to execute `IMPORT_REPLICA` as follows:

```
EXECUTE import_replica FOR content_object_id WITH store = 'name_of_target_storage_area', file = 'file_path_of_desired_file'
```

Setting proximity values for Content Server projection to a connection broker

Use the information in this section to help you set appropriate proximity values for Content Servers.

Guidelines

Use the following guidelines to define proximity values:

- Define proximity values for data servers in the range of 0 to 999. It is not necessary to include leading zeros.
- Define proximity values for remote content servers in the range of 9000 to 9999.
- Define proximity values for the primary server (the Content Server at the primary site) so that:
 - The server always projects the lowest proximity value among all servers projecting to any connection broker.
 - The server projects a content proximity value to the primary site connection broker that is lower than those values projected to the site by the remote servers.
 - The server projects content proximity values to remote connection brokers that are higher than those values projected from the remote sites.
- Define proximity values for a remote content server so that:
 - The proximity value it projects to a connection broker is higher than the proximity valued projected by the primary server.
 - The proximity value it projects to its local connection broker is lower than any projected to that connection broker from other sites.
- If two Content Servers have the same content proximity value and a client has established a data connection to one of them, the client uses the established connection for content requests also.
- A proximity value of -1 represents a server that is not available for any type of connection.

Example of selecting proximity values

The following table illustrates the proximity value guidelines. It shows projected proximity values for the servers in a repository called Demo. The repository has three sites: A, B, and C, with corresponding servers named DemoA, DemoB, and DemoC, and connection brokers named Connection BrokerA, Connection BrokerB, and Connection BrokerC.

Table 13. Example proximity values for a three-site configuration

Server	Projected Proximity Values		
	To Connection BrokerA	To Connection BrokerB	To Connection BrokerC
DemoA	0001	0002	0003

Server	Projected Proximity Values		
	To Connection BrokerA	To Connection BrokerB	To Connection BrokerC
DemoB	9002	9001	9003
DemoC	9003	9002	9001

At site A

When a user requests a connection, Connection BrokerA returns the following server and proximity value pairs:

Server	Proximity Value
DemoA	0001
DemoB	9002
DemoC	9003

For data requests, the client uses DemoA because it is the only server with a proximity value in the 0 - 999 range. (If there were more than one server with a proximity value in that range, the client would choose the server with the lowest value.)

For content requests, the client also selects DemoA also because it has the lowest projected content proximity (001). Using the same server for data and content at the primary site is fine. DemoA is the closest server for users at the primary site.

Note: You can set up both a data server and a remote content server at the primary site. The remote Content Server at the primary site must project a proximity value to the local connection broker that is higher than the data server's proximity value and a content proximity value that is lower than any value projected by the servers at the remote sites.

At site B

When a user requests a connection, Connection BrokerB returns the following server and proximity value pairs:

Server	Proximity Value
DemoA	0002
DemoB	9001
DemoC	9002

For data requests, the client uses DemoA, as it is the only server with a proximity value in the 0 - 999 range. For content requests, the client selects DemoB because its content proximity value (001) is lower than the content proximity value (002) projected by DemoA or DemoC.

At site C

When a user requests a connection, Connection BrokerC returns the following server and proximity value pairs:

Server	Proximity Value
DemoA	0003
DemoB	9003
DemoC	9001

For data requests, the client uses DemoA, the only server with a proximity value from 0 through 999. For content requests, the client selects DemoC because its content proximity value (001) is lower than the content proximity value projected by either DemoA or DemoB (003).

First-time use

The first time a remote content server issues a Getfile request to retrieve a document, the response time can be slow. The server constructs some internal structures to handle Getfile requests. Subsequent calls use the structures built with the first call and are faster.

Implementing multirepository models

This section describes how to set up the most commonly used building blocks for multi-repository distributed environments.

Repository configuration for distributed environment

The information in this section is a supplement to the installation and configuration information in *EMC Documentum Platform and Platform Extensions Installation Guide*. Use the information in this section to complete the configuration of your repositories if your repositories are participating in:

- A repository federation
- Object replication
- Distributed workflow

Connection broker setup

Every server in the distributed environment must be able to find the other servers through its connection broker.

Like an end user, a Content Server also uses a connection broker to find outside repositories. The connection brokers a server uses are identified in the `dfc.properties` file on the server host machine.

If the participating servers all project to the same connection broker and send to that connection broker for connection information, then no special setup is required. However, if the participating servers use different connection brokers for connection information, then additional configuration is required. In such cases, ensure that each participating server projects its connection information to the connection brokers used by all other participating servers.

Target connection brokers for server projection are defined in the server configuration object. The target information is stored in five repeating properties:

- `projection_targets`
The `projection_targets` property contains the name of the host machine on which the connection broker resides.
- `projection_port`
The `projection_ports` property contains the port number on which the connection broker is listening.
- `projection_proxval`
The `projection_proxval` property contains the proximity value that the server projects to the connection broker.
- `projection_enable`
The `projection_enable` property determines whether the server projects to the connection broker. If it is set to `TRUE`, the server projects to the connection broker. If it is set to `FALSE`, the server does not project the connection broker.
- `projection_notes`
The `projection_notes` property is a place for you to record short notes about the target. The property is 80 characters long.

Use the repository management facilities in Documentum Administrator to modify the server configuration object to set the projection targets for a server. The Documentum Administrator online help contains the instructions.

User setup

In the current implementation, the participating repositories must all use the same user name when establishing a connection for the internal jobs that manage distributed operations.

To configure the repository user account to use for distributed operations:

1. Create a user account that has Superuser privileges in each participating repository.
The simplest arrangement is to use the same user for all repositories, as this arrangement requires defining only a single global superuser account.
Note: If you are creating a global user through a federation's governing repository, the governing repository propagates the user to the other repositories. However, the user is created with no special privileges in the member repositories. Then connect to each member repository and set the user's privileges to Superuser.

2. In each participating repository, set the `operator_name` property of the server's server configuration object to the user's login name.
3. Set up the password for the user in each repository.

Password setup

The server for each repository must have a password for the user you defined. To define the password, set up a `dm_operator.cnt` for each repository and the appropriate number of `repository.cnt` files for each repository. The following procedures describe how to set up these files.

To set up the `dm_operator.cnt` file:

1. Create a file named `dm_operator.cnt` that contains the password for the user identified in the operator name.

The password must be the same in all the `dm_operator.cnt` files.

2. Place a copy of the file in each repository in one of the following directories:
 - Windows: `%DOCUMENTUM%\dba\config\repository_name`
 - UNIX and Linux: `$DOCUMENTUM/dba/config/repository_name`

To set up the `repository.cnt` file:

1. For each repository:
 - a. Create a file named `repository_name.cnt`, where `repository_name` is the name of the repository.
 - b. Put the user's password in that file.

The password can be the same or different for each repository.

2. In each repository, place a copy of the `repository_name.cnt` file for each of the other participating repositories in the following location:
 - Windows: `%DOCUMENTUM%\dba\config\repository_name`
 - UNIX and Linux: `$DOCUMENTUM/dba/config/repository_name`

For example, suppose there are three participating repositories: RepositoryA, RepositoryB, and RepositoryC. You create three files:

- `repositoryA.cnt`, which contains the user's password for Repository A
- `repositoryB.cnt`, which contains the user's password for Repository B
- `repositoryC.cnt`, which contains the user's password for Repository C

Then, you place:

- Copies of `repositoryB.cnt` and `repositoryC.cnt` in Repository A
- Copies of `repositoryA.cnt` and `repositoryC.cnt` in Repository B
- Copies of `repositoryA.cnt` and `repositoryB.cnt` in Repository C

Object replication jobs

If you have object replication jobs that have defined a user other than the user identified in the user setup process, errors can occur when you execute the jobs. The errors can occur because object replication jobs now use the `dm_operator.cnt` or `repository_name.cnt` password files to retrieve the password for the remote user.

For example, if you are running a replication job from target repositoryB that connects to repositoryA as a pull replication, the repositoryB server reads the local `dm_operator.cnt` or `repositoryA.cnt` file (depending on which is present) to obtain a password to use to connect to repositoryA.

To avoid errors, either give the remote superuser defined in your jobs the same password as the user identified in the user setup process or change the remote user in the job to the user identified in user setup process.

Distributed operations job activation

To complete the configuration for distributed operations, the `dm_DistOperations` must be activated in each participating repository. This job is installed in the inactive state.

Use the Jobs management facilities in Documentum Administrator to activate the job in the participating repositories. The `dm_DistOperations` job is categorized as a replication job.

Setting up a federation

A federation is two or more repositories that are bound together to facilitate management of global users, groups, and ACLs in a multirepository distributed configuration. One repository in the federation is defined as the governing repository. All changes to global users, groups, and external ACLs must be made through the governing repository.

If an enterprise includes multiple, mutually exclusive groups that do not share documents, you can set up multiple federations. However, a repository can belong to only one federation.

A federation can include repositories with trusted servers and repositories with non-trusted servers.

EMC does not recommend mixing production, test, and development repositories in one federation.

Choosing the governing repository

Consider creating a repository to be the governing repository. Such a repository has the following advantages:

- Small size
- Easier backups
- Supporting jobs are run in a small repository

If you are creating a federation from a group of existing repositories, choose the dominant repository as the governing repository. The dominant repository is the repository in which most users are already defined as repository users.

Identifying user subtypes for propagation

By default, the federation jobs propagate all global users defined as `dm_user` objects in the governing repository. If you have created subtypes of the `dm_user` object type, global users defined with those subtypes are not propagated automatically.

To propagate users defined by `dm_user` subtypes, the subtypes must be present in all member repositories. The subtypes also must identify the user subtypes to propagate when you create the federation. You can add or delete from the subtypes after the federation is created also.

Creating a federation

Federations are created and managed through Documentum Administrator. The following procedure summarizes the necessary steps. The Documentum Administrator online help contains the detailed instructions on each step.

To create a federation:

1. Decide which repositories participate in the federation and which repository is the governing repository.
2. Make sure that the governing repository is projecting to the connection brokers for all member repositories.
3. Make sure that the member repositories are projecting to the governing repository's connection broker.

Only repositories that project to the governing repository's connection broker appear in the list of possible member repositories.

Projection targets are defined in the server configuration object, which can be modified using Documentum Administrator. The Documentum Administrator online help contains the instructions on adding connection broker projection targets to servers.

4. Create the federation using Documentum Administrator.

The Documentum Administrator online help contains the instructions.

Guidelines:

- Checking the checkbox to make all current users and groups in the repository global activates the `dm_FederationUpdate` job. This job is responsible for starting the methods that keep the global objects synchronized across the federation. The job is inactive when installed.

If you do not check the checkbox to set all users and groups to global, manually set the `dm_FederationUpdate` job to active. Use Documentum Administrator to do that.

- The federation name must consist of ASCII characters and cannot contain any single quotes ('). You cannot change this name after the federation is created.

- The server uses the superuser name and password you provide for each member to connect to the member repository when running federation jobs and methods that update the global objects.
- If the member repository is running on a domain-required Windows server or if the superuser is in a domain that is different from the default, enter a domain for the superuser.

Implementing object replication

The following procedure lists the basic steps for planning and implementing object replication.

To implement object replication:

1. Define the business requirements for replication.
2. Determine whether your current infrastructure meets the needs of the business requirements.
3. Determine the needed computing resources.
4. Set up each site.
5. Define the replication jobs.

Defining business requirements

Begin planning object replication implementation by determining your business requirements for replication. Two considerations can affect your business requirements:

- A replication job can only replicate documents to one target repository from one source repository.
- A replication job does not replicate users, groups, or object types.

The first consideration means that more than one replication job can be required to satisfy a business requirement. For instance, to distribute documents to three geographically dispersed locations, you need three replication jobs.

The second consideration means that you must coordinate multi-site, multirepository users, groups, and access permissions as a business function between repositories. Object types, users, and groups, are not replicated as part of the replication job. ACLs (dm_acl objects) can be replicated, depending on how you configure the job.

Setting up a repository federation that includes all repositories participating in object replication enables you to coordinate users, groups, and security access across repositories. In a federation, users, groups, and external ACLs are global objects. You can change them at the governing site and propagate the changes automatically to other members of the federation. Manually manage object types and formats regardless of whether the participating repositories are in the same federation.

If you choose not to use a federation, manage users, groups, and security manually.

For example, suppose a planning group has users in two different repositories and the repositories are not members of the same federation. Tom, Dick, and Harry are users in repository 1, while Jane

and Mary are users in repository 2. To allow these users to share documents of the object type `planning_doc`, the administrators for the two repositories must, at a minimum:

- Ensure that the `planning_doc` object type is defined identically in both repositories

Note: If you replicate an object into a repository in which the object's type does not exist, the operation creates the type in the target repository. However, if the type exists in both source and target repositories, define it identically in each.

- Create a planning group within each repository
- Add the users to both planning groups
- Create an ACL with an entry for the planning group that gives the group sufficient access to view replicated documents and their annotations
- Create a folder appropriate for the planning group's document sharing

The following sections describe the typical business requirements for users, groups, object types, security, workflow, and administration needs. These sections also discuss how these requirements affect your replication implementation.

Functional divisions and groups

If your company has clearly defined functional divisions, translate these divisions into appropriate Documentum group objects, to be defined in all repositories involved in replication. If you are developing an enterprise-wide replication plan before actually creating any repositories, create a standard script that defines these groups.

If you are not placing the repositories in a federation, you can run the standard script in each repository. This practice ensures that the group definitions meet the business requirement and are standard across all repositories. Edit the script for each repository so that the group creation statements are specific to the users in that repository.

If the participating repositories belong to a federation, it is only necessary to run the script at the governing repository site. The groups are automatically propagated to the other sites when the federation is created and the management jobs are active.

If you are converting existing repositories to accommodate a new or modified business requirement for replication, you might find that you must create new groups or modify existing groups. It still could be possible to utilize a standard script for this purpose, but some repository-by-repository modifications could also be necessary.

Document types

Document types usually evolve out of a combination of enterprise-wide and functional business requirements. The document types must have properties that capture all of the information necessary for users to access and utilize the document in all business contexts. To preserve this information in replicated documents, it is important to define and maintain enterprise-wide document type definitions.

Document types can be defined using a standard script, which is recommended if your repositories are not yet created. If you are converting existing repositories to meet replication business requirements, you might have to create new definitions or modify existing definitions.

Maintaining enterprise-wide document type definitions is desirable. It preserves all property and content information when a document is replicated. However, it is not mandatory for Documentum replication. If the definitions are not identical, the system copies all information possible and ignore any information that cannot be replicated.

For example, suppose the user-defined document type `planning_doc` has 15 user-defined properties including the property `project_leader` in repository 1. But it lacks that property in repository 2. Replication from repository 1 to repository 2 would result in a replica with 14 of the user-defined properties, but not `project_leader`. A replication from repository 2 to repository 1 would result in a replica with all 15 properties present, but with no information in the `project_leader` property.

If you replicate an object whose type is not defined in the target repository, the operation creates the type in the target repository as part of the replication process.

User distribution and geography

Documentum defines users at the repository level. If a user is defined in more than one repository, that user has a unique Documentum user ID in each repository. In a default installation, the server considers each a different user, even if the `user_os_name` and `user_name` properties are identical in both repositories. In a federation, users are global objects managed by the governing repository. The server considers users having the same `user_os_name` and `user_name` properties in different repositories to be identical.

There are no user management concerns if you are replicating between repositories in the same federation. The federation's management jobs ensure that the user definitions are the same in each member repository.

Replication between repositories that are not in a federation or not in the same federation does not require user definitions to be the same across the repositories. In such cases, the replication job maps the ownership of the replicated objects to users in the target repository.

If your company has no policy for uniquely identifying users across sites, you can define users repository by repository. If there is a policy for uniquely identifying users across all sites, you can use this identification scheme in Documentum without modification. The product works in either situation.

When a new user joins a project or the company, it is up to the administrator to add the user to the appropriate groups. These groups include any groups participating in replication processes if necessary. Generally, cross-repository coordination is not required. However, you might have set up some customizations that require coordination. For example, if you create registered tables of remote repository users and `user_names` in each repository (to support cross-repository event notification, for example), adding a new user to a repository requires coordination. In such cases, the new user must be added to the registered tables in the remote repositories also.

Security

The choices you make for security depend on the replication mode you are using. There are two replication modes: `nonfederated` and `federated`. Each provides different security options. describes the security options available for each mode.

If you are using a nonfederated mode and choose to assign the same ACL to all replicas, security requirements can dictate which documents are included in a replication job. You could implement a replication business requirement with complex security by creating an ACL containing grants to many groups within the target repository, each with different access rights. Alternatively, the replication job could be divided into a group of replication jobs, each with its own simple but unique ACL.

Infrastructure

After you define your business requirements for replication, determine whether your infrastructure meets the needs of the requirements. Infrastructure is defined as the hardware, networking software, and people that support Documentum replication. Determine the following:

- Whether you have adequate hardware resources
- Whether you want to perform replication online or offline
- How you want to assign the duties associated with managing a replication site.

The following sections address these issues.

Reference metrics

Reference metrics provide a baseline for capacity planning, identifying potential infrastructure weaknesses, and determining what performance can be expected. To help determine whether the hardware and software infrastructure at your site is adequate to support your replication needs, compute reference metrics on each server that participates in your replication configuration. Compute the metrics during both off-peak and peak times because replication jobs can run at both times.

Obtain the following baseline metrics:

- Server CPU capability

Record the elapsed time for a local Documentum client to create and save 1,000 objects that have no content. This operation is an approximating metric for gauging CPU capability. Comparing CPU capability across environments is difficult.

- Server disk capacity

Report `df` command information for the UNIX and Linux servers.

- Server disk speed

Perform a local Documentum client `Setfile/Getfile` and record the elapsed time for each.

- Network speed

Perform a `Setfile/Getfile` between each server pair participating in replication and record the elapsed time for each.

Note: Documentum provides a `Docbasic` script that you can use as a base for writing a script to obtain all of the reference metrics except disk capacity. You can find the script in `%DM_HOME%\unsupported\replicate\metric.ebs` (Windows) or `$DM_HOME/unsupported/replicate/metric.ebs` (UNIX and Linux).

For example, the following table shows the baseline metrics obtained for the two servers participating in replication testing. The metrics are expressed as minutes.

Metric	Fox, off-peak	Fox, peak	Bison, off-peak	Bison, peak
Server CPU	3:40	8:25	3:50	7:14
Local Setfile	0:30	1:30	:17	1:25
Local Getfile	0:17	1:57	0:14	1:30
Remote Setfile	31:02	37:29	30:56	36:17
Remote Getfile	29:57	35:56	29:30	36:49

If you obtain inadequate metrics in any of these areas, factor that into your infrastructure planning:

- Alleviate CPU shortfalls by adding processors to your server or by putting RDBMS processing on a different server than Content Server.
- Alleviate disk shortfalls by procuring additional disk devices and rearranging the map of logical devices to controllers and physical disks.
- Alleviate network shortfalls by examining network router losses, procuring additional network bandwidth, or both.

If you cannot resolve network bandwidth shortfalls, consider using the offline replication option.

Network replication options

There are two basic options for object replication: online and offline.

Online replication is the default. In online replication, the replication job originates at the target site and performs the following tasks:

- Synchronously requests source-site processing
- Synchronously transfers the resulting dump file
- Synchronously performs the target-site processing to complete the replication

In offline replication, the replication job originates at the target site and requests the source-site processing. Asynchronously, the source site places the dump file in a requested location. A system administrator then uses ftp or tape to move the dump file from its source location to the target location. After the dump file is placed in its target location, the replication job picks up where it left off, performing the remaining target-site processing.

Replication system administration

In addition to the initial planning and coordination between sites, enterprise-wide replication has ongoing administrative tasks. For example, the duties associated with replication can include:

- Adding groups and users to the appropriate repositories
- Resolving problems when networks or repositories are not available
- Monitoring the progress of replication jobs and resolving failures
- Coordinating object type definitions across repositories

An enterprise with a number of large repositories at relatively autonomous sites can require one additional full-time individual per site to administer replication. This individual requires conventional Documentum skills and business knowledge to translate business replication requests into appropriate replication jobs.

Note: Auditing is not supported when performing refresh operations on a replica. Performing an audit each time a refresh operation is performed on a replica would affect performance and use up disk space.

Determining computing resources

Determining the computing resources required for replication is primarily a matter of:

- Listing business requirements
- Translating those requirements into specific replication jobs
- Extrapolating the required machine resources based on the parameters of each job

Disk space requirements and job scheduling at each site require examination. This section demonstrates this process by example.

Determining needed jobs

XYZ Enterprises has three geographically dispersed repositories: X, Y, and Z. They have two products on the market that were developed at their original site, X, which continues to control everything related to these products. However, Site Y needs rapid access to the documentation for products 1 and 2, and site Z needs rapid access to product 2 documentation.

Additionally, XYZ Enterprises is developing a new product that requires collaboration among all three sites. Every document produced with the new product is replicated from its originating site to the other two. All three sites are expected to generate large amounts of review annotations, which must also be replicated to all sites.

The company has defined the following six replication jobs to achieve these business objectives:

Example 4-1. Product 1 Replication Jobs:

- Job 1: From X to Y once a week

Example 4-2. Product 2 Replication Jobs:

- Job 2: From X to Y once a week
- Job 3: From Y to Z (X documents received indirectly from Y) once a week

Example 4-3. New Product Replication Jobs:

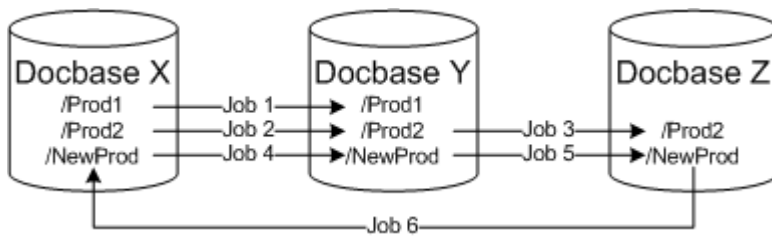
- Job 4: From X to Y every 2 hours
- Job 5: From Y to Z every 2 hours
- Job 6: From Z to X every 2 hours

Jobs 1 and 2 represent the standard distribution of X documents to Site Y. Because it is possible to replicate replica documents, Job 3 completes the distribution by replicating X documents to Z through Y.

For Jobs 4, 5, and 6, each site's target and source folder for the new product documentation is called /NewProd. This configuration and the circular nature of Jobs 4-6 mean that each site has its own documents and annotations; in addition, all documents and annotations from the /NewProd folders of the other sites in its own /NewProd folder.

The following illustration depicts these jobs.

Figure 17. XYZ jobs



Disk space requirements

This section illustrates how to estimate the disk space required for replication as accurately as possible. The values used in this section for each required estimate are only examples. For your calculations, determine the correct figures for the required estimates.

For replicated documents

Use a table to calculate disk space requirements. First, compute the requirements for the source repository documents and metadata. This operation is a function of the:

- Number of source documents and virtual descendant (related) documents
- Estimated average document size
- Estimated rendition size
- Estimated annotation size

Most documents have multiple versions, so be sure to take versions into account also.

This table shows these figures for the XYZ Enterprises example.

	Product 1, X	Product 2, X	New Product, X	New Product, Y	New Product, Z
Number of documents	1,000	2,000	100	50	75
Number of versions	6	5	4	3	2
Content (KB)	10	20	5	15	18

	Product 1, X	Product 2, X	New Product, X	New Product, Y	New Product, Z
Renditions (KB)	20	40	5	15	18
Annotations	0	0	33	5	5
Total content (KB)	30	60	43	35	41
Estimated Total (MB)	200	650	18	4.35	11.41

Total content is the sum of content, renditions, and annotations for each document.

The total represents the total size of the repository document and metadata content. It is the product of the total number of documents (number of documents times the number of versions per document) and the total number of bytes per document (total content + 2,500 bytes per document for metadata overhead), or

Total = (no. of documents x no. of versions per document) x (total content + 2,500 bytes)

The metadata overhead varies, depending on the complexity of the document types.

After you calculate source disk space, you can project the total requirement to each of the sites.

The total requirement is the sum of the source and replicated documents at each site plus required temporary space for dump files.

Temporary space for dump files

Each replication site needs temporary space for dump files. describes the storage locations of the source and target dump files. Replication processing generates a dump file at the source site. It then stores it as the content of a document in the source repository before transferring it to the target repository. After the dump file arrives at the target site, it is filtered into another dump file specific to the target repository. Owner and ACL information for each document are modified, if necessary, to conform to replication job requirements, and some objects are removed. The temporary disk space available for dump files on the source and target sites must be twice the size of the largest dump files.

If disk space is limited on either the source or target site, consider running the job as a series of smaller dump and load operations.

If you are not planning to run the job using multiple dump and load operations, calculate temporary space. When doing so, assume that at some point a full refresh replication can occur. A full refresh replicates all documents in a single replication request rather than incrementally as they are modified.) Increase the temporary disk space if scheduling requires you to run multiple replication jobs simultaneously. The following table shows the calculations for XYZ Enterprises. The Temp column represents twice the disk space requirement of the largest full-refresh replication job at that site. In this example, Product 2 has the largest content size, so that figure is doubled for the Temp calculation.

Site	Product 1	Product 2	New Product	Storage	Temp	Total
X	200	650	22.76	872.76	1300	2,172.76

Site	Product 1	Product 2	New Product	Storage	Temp	Total
Y	200	650	22.76	872.76	1300	2,172.76
Z	-	650	22.76	672.76	1300	1,972.76

Job scheduling

A replication job consists of three components:

- Source site processing to generate the dump file
- Dump file transfer
- Target site processing to load the dump file

The processing time required to complete all three steps is the processing window for the job. To verify that your replication schedules are achievable, estimate the processing window for each job.

In the XYZ Enterprises example, Jobs 1, 2, and 3 can be scheduled for off-peak server usage times because they are run weekly. It is unlikely that these jobs will conflict with other uses of the servers. On the other hand, Jobs 4, 5, and 6 are scheduled to run every two hours. These jobs are running during peak load times for the source and target servers. If the processing window exceeds two hours, the schedule cannot be maintained.

To assess this possibility, develop an estimate of elapsed time for the three steps, based on reference metrics and the size of the replication job. Use the CPU metric and the local Setfile metric to estimate the time required for dump file generation at a source. Use the remote Getfile metric to estimate the time required for dump file transfer. Use the CPU metric to estimate the time required by the filter program and dump file load. The sum of these estimates provides an estimate for total processing time.

The following table shows the calculations for Job 4, using the peak reference metrics shown in sample table for reference metrics, and assuming a full refresh of the replicated objects. The metrics measuring documents processed per second are obtained by dividing the size of the server executable by the operation's elapsed time. The server executable was the file used to generate the metrics. Peak values are used because Job 4 runs throughout the day.

Processing Step	Reference Metric	No. of Objects or File Size	Time
Generate dump file	0.5 docs/second	700	350 seconds
Store dump file	111K/second	22.76 MB	205 seconds
Transfer dump file	4.4K/second	22.76 MB	5172 seconds
Filter dump file	0.43 docs/second	700	301 seconds
Load dump file	0.43 docs/second	550	237 seconds
Total			6265 seconds (1 hr, 44 min, 25 sec)

Handling overlapping jobs

Overlapping jobs are two replication jobs that replicate the same object either into one repository or into and out of the same repository. For example, suppose you have two replication jobs with the same source and target repositories. If the execution times overlap and the jobs happen to replicate some of the same objects, errors can occur. Or, suppose one job A replicates into a target repository YY and another job B that replicates from repository YY. In such cases, if the execution times overlap, it is possible that job B attempts to replicate objects that job A is loading. This operation is an error.

The recommended way to avoid such clashes is to group jobs that can overlap into a job sequence. A job sequence is a set of jobs that are executed in a user-defined order. If a job in the sequence has a predecessor in the sequence, that job is not executed until its predecessor completes successfully. The Documentum Administrator online help contains more information about job sequences.

Site setup

Establishing and following enterprise-wide standards for groups, document types, and security is critical to the success of replication.

Some other aspects of an installation must also be reviewed and any required reconfiguration performed before you start replication jobs. The following sections describe these considerations.

Connection broker setup and validation

To use all the features associated with object replication, each participating site must project to the connection broker at each of the other participating sites. However, if you want only to create read-only replicas in the target repository, then cross-projection is not required.

This section describes how to set up cross projection. It can be helpful to have a listing of the repositories already known to your local connection broker before setting up cross projection.

To add a new repository to a cross-projection environment:

1. Modify the server configuration object for the new repository to add connection broker projection targets for each remote site that is a potential replication target.

After you modify the server configuration object, reinitialize the server for the changes to take effect. Use Documentum Administrator to modify the server configuration object. The Documentum Administrator online help contains the instructions.

2. Modify the server configuration objects in existing repositories that are potential replication sources to add a connection broker projection target for the new repository.

After you modify the server configuration object, reinitialize the server for the changes to take effect. Use Documentum Administrator to modify the server configuration object. The Documentum Administrator online help contains the instructions.

Macintosh access protocol

If you intend to replicate Macintosh documents, all participating repositories must be using the same Macintosh access protocol. The protocol controls header, trailing, and type creator information that is processed to and from the resource fork when the client DFC is processing the resource fork to and from the server.

The access protocol is set in the `mac_access_protocol` property of the repository's configuration object. Although the replication setup scripts check for consistency, confirm that your sites are consistent before the installation. To determine which protocol a repository is using, execute the following DQL query:

```
SELECT "mac_access_protocol" FROM "dm_docbase_config"
```

Disk space for temporary files

Replication creates temporary files in both the source and target repositories. The largest temporary files are the dump files. The temporary files are stored in the `%DOCUMENTUM%\share\temp\replicate` (Windows) `$DOCUMENTUM/share/temp/replicate` (UNIX and Linux) directory in each installation. The replication operation creates a directory underneath this directory for each repository participating in replication on that machine. Make sure that this directory is located on a file system with enough space for the temporary files.

On UNIX and Linux, if the directory is not located on a file system with enough space for these temporary replication-related files, you can allocate a separate area and create a UNIX and Linux link, called `replicate`, to the `share/temp` area. For example:

```
% cd /u107/dm/* file system with adequate space */
% mkdir dntemp % cd $DOCUMENTUM/share/temp
% ln -s /u107/dm/dntemp replicate
% cd replicate <create directories for all repositories on site that will participate
in replication and set all permissions for the installation owner>
```

Content storage

Disk capacity for replication is a critical issue.

On the source side, the documents associated with the replication dump files are stored in a storage area called `replicate_temp_store`. This area is created by default in the same directory as the default SysObject file store area (`filestore_01`).

Where content is stored in the target repository is dependent on the replication mode chosen for the job.

Cabinets and folders

Object replication jobs replicate objects linked to a particular folder or cabinet called the source folder. The source folder is identified in the job definition. The replicated objects are stored in the target repository in a particular folder or cabinet designated as the target folder in the job definition. Both

the source folder and the target folder must exist before you define the replication job. When you create these folders (or cabinets), keep in mind the following code page requirements:

- If the participating repositories have the same `server_os_codepage` value, the names of the source and target folders must be compatible with that code page.
- If the participating repositories have different `server_os_codepage` values, the names of the source and target folders must consist of only ASCII characters.

The replication process does not replicate the source folder or cabinet in the target repository's destination folder or cabinet. However, the process does replicate the folder hierarchy found within the source folder or cabinet.

If the source folder or cabinet contains a reference link, the replication process replicates the reference link but does not replicate any objects contained within the object pointed to by the reference link.

If two repositories are exchanging documents using folders with the same names and subfolder structures, name the subfolders identically in both repositories. For example, suppose that repository1 and repository2 both have `/Product` folders that are used for replication. In repository1, `/Product` has a subfolder named `market_outlook`, while in repository2, `/Product` has a sub folder named `marketoutlook`. The names of the subfolders only differ by one character, the underscore. However, because the names do not match exactly, after bidirectional replication both repositories have the following folder paths:

- repository1
 - `/Product/market_outlook`, which contains source documents
 - `/Product/marketoutlook`, which contains replicated documents
- repository2
 - `/Product/market_outlook`, which contains replicated documents
 - `/Product/marketoutlook`, which contains source documents

This operation was not the desired result. If subfolders are named consistently, all documents, whether replicas or not, appear in one folder in both the source and target repositories.

Additionally, make sure that all users participating in the replication have permission to link documents into the source folder.

Defining jobs

New jobs are defined using Documentum Administrator. You must be a superuser to define a replication job.

Before you create a job, the target folder or cabinet for that job must exist. The server only allows you to specify an existing folder or cabinet as the target when you are defining a job. If the folder or cabinet does not exist when you define the job, save the unfinished job definition. Then create the folder or cabinet and finish the job definition.

Guidelines for all jobs

Observe the following guidelines when creating a job:

- To define a push replication job, create the job in the source repository. To define a pull replication job, create the job in the target repository. If you want the job run by a mediator repository, create the job in the mediator repository.
- If the underlying databases of the repositories are using different code pages, set the Codepage field for the replication job to UTF8. If the databases are using the same code page, set the field to that code page.
- Choosing fast replication can make the job run faster. However, it accomplishes the faster speed by limiting which related objects are replicated.
- If you define multiple jobs with the same target and source repositories, all jobs on the source repository must run under the same user account.
- If the job fails, you can set a flag in the job definition that directs the server to restart the replication job. However, we recommend that you make sure that your job is correctly defined and running smoothly before you set that flag. The restart feature is intended to recover jobs that fail because some factor in their environment, such as the source or target repository, is unavailable.

Guidelines for multidump file jobs

Observe the following guidelines when defining a job that uses multiple dump and load operations to perform the replication:

- Set the `-objects_per_transfer` argument in the method arguments to number of objects you want to transfer in each operation.

It is recommended that you do not specify fewer than 1000 objects in the argument.

- You must define the job as a fast replication job.
- You cannot define the job as a manual transfer job.

Setting up tracing

Setting the trace level to any value except 0 turns on tracing for the replication job. Setting the argument to 0 turns off tracing for the job.

All tracing information appears in one file whether you are using one dump and load operation or multiple dump and load operations for the job.

Manual dump file transfers

If you manually transfer the replication job dump file to the target repository, place the dump file in the following location on the target repository:

- Windows:


```
%DOCUMENTUM%\share\temp\replicate\target_db_name
```

- UNIX and Linux:

```
$DOCUMENTUM/share/temp/replicate/target_db_name
```

When you put the dump file in this location, be sure to give the file the same name as the original dump file. Within 60 minutes, the replication job agent detects the presence of the file and automatically loads the file.

Note: If the replication job is using multiple dump and load operations, you cannot use a manual file transfer.

Best practices for object replication

Observe the following rules regarding object replication to preserve structural and data consistency in the participating repositories:

1. If a folder is a target folder in one replication job and a source folder in another replication job, do not schedule the two jobs to run concurrently or overlapping.
2. Do not change job settings for the source and target after a job has been run.
3. If the same documents are replicated to multiple target folders in the same repository, the parameters of each job must match and the jobs must not run concurrently or overlapping. Version mismatch errors can occur if the jobs are run concurrently.
4. To delete a replica from a target repository, first move the source object out of the source folder. If there are multiple jobs replicating the object to that target repository, remove the object from each source folder that replicates to that target repository. Then, use a Prune method and delete the replica's entire version tree in the target repository.
5. Do not create local folders in or link local folders to a target folder.
6. Do not move a replicated folder in a target repository to another location.
It is acceptable to link a replicated folder to another, local folder or cabinet. However, do not remove the primary link to the target folder.
7. To delete a folder that contains replicas, first remove the replicas from the folder.

Managing single-repository models

This section contains procedures for administering the building blocks of a single-repository distributed configuration.

Adding a distributed component

Use this procedure if you are adding a new remote site to an existing single-repository distributed installation.

To add new site to a distributed storage area installation:

1. Make sure that the installation at the new site complies with the guidelines
2. Make sure that there is enough disk space at the new site to support the content storage requirements of a distributed storage area.
3. At the new site:
 - a. Install a remote content server and ACS server.

Installing Content Server and the ACS server creates the required server configuration object for the new servers. The ACS configuration object updates the `acs.properties` file for the new ACS server and creates a file store storage area at the remote site. *EMC Documentum Platform and Platform Extensions Installation Guide* contains more information.
 - b. Start the remote site's server.
 - c. Add the file store storage area at the remote site to the distributed store storage area as a component.
 - d. Add the new component to the **Far Stores** list in the server configuration object for each of the other component sites in the distributed storage area.
 - e. Optionally, reset the proximity values for the remote content server.

The installation process automatically sets a remote server's proximity to its local connection broker as 9001 and its proximity to the primary Content Server as 9010. You can reset the value to reflect your network's topography.
4. Ensure that users and groups accessing distributed documents from the new site are authenticated by servers at all sites, using the same mechanism.

In a configuration using a distributed storage area, when a user connects to a repository, the client sends two connection requests: one to the data server and one to the remote content server. Each server must be able to authenticate the user's name and password using the same authentication mechanism.
5. Ensure that the machine hosting the Content Server is using UTC time and is synchronized with the other machines hosting distributed components.

Servers at the remote sites use global login tickets configured to be valid for 5 minutes to connect to the primary site to fetch content. If the clocks in the host machines are not synchronized, the connection can fail if the host machine finds the login ticket invalid if the time difference is greater than 5 minutes.
6. To set up repository access for web-based clients at the remote site:
 - a. Define at least one network location for the ACS server at the site.
 - b. Add the network location to the ACS configuration object.

Add the location to the ACS configuration object for each ACS server that can service requests from that network location.
 - c. Define the network location's proximity to the ACS server or servers.
 - d. Define the ACS server's projection to a connection broker.

Removing a distributed component

It is possible to remove a site from a distributed storage area. Removing a site does not destroy the underlying directory or the files stored at the site. Use the content storage facilities of Documentum Administrator to remove a component from a distributed storage area.

Removing files from component storage areas

When a user deletes a document from the repository, regularly scheduled repository maintenance removes the associated files from the storage areas. You can remove a file from a component storage area without removing the copies in other component areas and without destroying the associated SysObject, content object, or replica record object.

To remove a file without removing the associated objects, use the DELETE_REPLICA administration method. DELETE_REPLICA removes a file from a storage area and modifies the replica record object associated with the file's content object. If you remove the file using operating system commands, the replica record still indicates that the file exists in the storage area. This operation can cause errors when users try to access the file.

DELETE_REPLICA has one argument, STORE. This string argument identifies the storage area that contains the file you want to remove.

Using Documentum Administrator is the recommended way to execute DELETE_REPLICA. However, you can also use DQL EXECUTE or an apply method.

Using DQL EXECUTE

If you use DQL EXECUTE, the syntax is:

```
EXECUTE delete_replica FOR content_object_id WITH store = 'storage_area_name'
```

Specify the object ID of the file's associated content object. *storage_area_name* is the name of the storage object for the component storage area from which you are removing the file.

Troubleshooting surrogate get

The surrogate get feature provides automatic on-demand content replication within the components of a distributed storage area. It is intended for configurations that are not using shared directories.

Tracing surrogate get

You can use the trace file to trace dm_SurrogateGet method operations. You can also turn on Trace Launch to trace the method's invocations.

The trace file

If surrogate tracing is turned on, all dm_SurrogateGet method operations and timing information are traced in a file called sugg.trc. The file is stored in %DOCUMENTUM%\bin (Windows) or \$DOCUMENTUM/bin (UNIX).

Turning on trace file generation

The sugg.trc file is not generated by default. Manually turn on tracing for SurrogateGet.

To trace surrogate get operations:

1. Open the mthd6.ebs script in a text editor.
The script is typically stored in %DM_HOME%\install\admin (Windows) or \$DM_HOME/install/admin (UNIX).
2. Remove the single quote (') at the beginning of the following line:

```
'ret% = dmAPIExec("trace,s0,10,sugg.trc")
```
3. Save your changes and exit the editor.

Tracing method invocations

To trace method invocations for surrogate get, set the Trace Launch property on the property page for the job in Documentum Administrator.

After you turn on Trace Launch for the method, the system logs dm_SurrogateGet method's command line in the server log file whenever the method is invoked.

Resolving problems

If the on-demand content replication of the SurrogateGet method is not working properly, check the following configuration items:

- The only_fetch_close property
- The get_method properties
- Connection broker projection targets
- Time settings on the participating host machines

The only_fetch_close property

The only_fetch_close property is defined for the distributed store object. It controls whether a server invokes the surrogate get method to fetch content files from component storage areas defined as far for the server.

Make sure that this property is set to TRUE. When the property is FALSE, the surrogate get method is not invoked.

If Trace Launch is turned on for SurrogateGet, you can view the server log file to determine if the surrogate get method is invoked when the server fetches a content file stored in a far storage area. The method's command line is recorded in the server log file when the method is invoked.

The get_method properties

The server invokes the method defined in the get_method property for the distributed store object and the component storage areas. All must contain dm_SurrogateGet if you are using the SurrogateGet tool or the name of your user-defined surrogate get method. Check the property's value and modify it if necessary in the distributed store object and in the component storage areas.

You can use Documentum Administrator to check the property's value.

Connection broker projection targets

Ensure that the server at the site where the content file is stored is projecting correct connection information to the connection broker used by the server invoking the SurrogateGet method.

The connection broker used by the server invoking SurrogateGet must know about the server at the site where the content file is stored. This means that the server at each site in a single-repository distributed configuration must project its connection information to the connection broker at each of the other sites in the configuration.

Time settings

If the job is failing, ensure that the host machine on which the job runs and the host machine of the Content Server to which the job is connecting are using UTC time. Also ensure that the clocks are synchronized.

Surrogate get uses a login ticket to connect to servers to obtain content files. The login tickets are valid for 5 minutes. If the machine on which the dm_SurrogateGet method is running and the host machine of the server to which it connecting are not synchronized in time, the ticket can be invalid (timed out) on the target due to perceived differences in time. In such cases, the job fails.

Overriding remote content server use

Remote content servers improve repository query performance for desktop users at remote sites. However, you can override this functionality on occasion. For example, most the administration methods that manipulate content files and storage areas require the client to use one server for both content and data requests.

There are two ways to override the remote content server functionality:

- Set the `dfc.content.use_content_server` key in the `dfc.properties` file to `FALSE`
- Specify a server when you issue the connection request

Using the `use_content_server` key

The `dfc.content.use_content_server` key is an optional key in the `dfc.properties` file. If you set the key to `FALSE`, desktop clients always connect to one server for both data and content requests. The key is `TRUE` by default.

With one exception, `TRUE` causes the client to connect to two different servers for data and content requests. The exception occurs when the connection request specifies the local server.

Using the connection request

DFC allows you to specify a repository using syntax that identifies which server to use for connection. The server configuration object identifies the name of the server.

The server is identified in the connection request. The desktop client application use that server for content and data requests when `dfc.content.use_content_server` is `NULL` or `FALSE`. When `dfc.content.use_content_server` is `TRUE`, the server identified in the connection request is used instead of the remote content server only if it is the user's local server.

Using both `use_content_server` key and the connection request

To illustrate how the `dfc.content.use_content_server` key setting interacts with a server specification in a connection request, assume that an enterprise has two servers, `cat` and `mouse`, installed on different machines and that:

- `cat`'s proximity to the connection broker is defined as 2.
- `mouse`'s proximity to the connection broker is defined as 9001

The following table describes the interaction between the `use_content_server` setting and a server specification in the connection request.

Connection request	Data server	Remote content server	Explanation
<code>use_content_server</code> is not set			

Connection request	Data server	Remote content server	Explanation
connect, <i>repositoryname</i> , <i>username,password</i>	cat	mouse	The connection request does not identify a server and <code>use_content_server</code> is not set, so the client connects to <code>cat</code> as the data server and <code>mouse</code> as the remote content server.
connect, <i>repositoryname</i> .cat, <i>username,password</i>	cat	cat	The connection request identifies a server and <code>use_content_server</code> is not set to TRUE, so the client uses <code>cat</code> for both data and content requests.
connect, <i>repositoryname.mouse</i> <i>username,password</i>	mouse	mouse	The connection request identifies a server and <code>use_content_server</code> is not set to TRUE, so the client uses <code>mouse</code> for both data and content requests.
<code>use_content_server=TRUE</code>			
connect, <i>repositoryname</i> , <i>username,password</i>	cat	mouse	The connection request does not identify a server and <code>use_content_server</code> is set to TRUE, which reinforces the default behavior. The client connects to <code>cat</code> as the data server and <code>mouse</code> as the remote content server.
connect, <i>repositoryname</i> .cat, <i>username,password</i>	cat	mouse	The connection request specifies a server for the connection. However, the client connects to two different servers because the <code>use_content_server</code> key is TRUE.

Connection request	Data server	Remote content server	Explanation
connect, <i>repositoryname.mouse</i> <i>username,password</i>	mouse	mouse	The connection request identifies the local server, which is also the default remote content server. Consequently, the client uses mouse for both data and content requests.
use_content_server=FALSE			
connect, <i>repositoryname</i> , <i>username,password</i>	mouse	mouse	The connection request does not identify a server. Because use_content_server is FALSE, the client does not have to connect to different servers for data and content requests. Consequently, the client connects to the local server for both data and content requests.
connect, <i>repositoryname</i> <i>.cat, username,password</i>	cat	cat	The connection request identifies a server. The client connects to the server for both data and content requests because use_content_server is FALSE, so there is no requirement to use different servers.
connect, <i>repositoryname.mouse</i> <i>username,password</i>	mouse	mouse	The connection request identifies a server. The client connects to mouse for both data and content requests because use_content_server is FALSE, so there is no requirement to use different servers.

Login failures in remote content server setups

When remote content servers are in an installation, remote clients connect to both a data server and a remote content server by default when a session starts. Each connection requires a separate user authentication.

If a login failure occurs at the data server site, the failure occurs when the user tries to log in to the repository.

If a login failure occurs at the remote content server site, the user is logged in to the repository, but all content transfer requests fail.

Tuning query performance

As users add documents to a distributed storage area, the `dmi_replica_record` tables in the repository grow. This growth affects the performance of queries against the documents in the distributed storage area. To ensure the best performance, make sure that you are updating the statistics for the RDBMS tables underlying the repository. The query optimizer uses statistics to choose the best query plan.

You can generate statistics using either the Documentum Update Statistics system administration tool or the statistics tool provided by the underlying RDBMS. Update Statistics generates statistics for all tables in the repository. The RDBMS tools typically allow you to specify the table or tables for which you want to generate statistics.

The Documentum Administrator online help contains more information about Update Statistics. Consult the documentation provided by your RDBMS vendor for information about running statistics directly in the RDBMS.

Managing multirepository models

This section describes how to perform common administrative tasks for multirepository distributed configurations.

Manipulating a federation

This section contains procedures for managing a federation.

Adding a member

A repository can belong to only one federation. Consequently, the member you are adding cannot currently belong to any federation.

Use the federation management facilities of Documentum Administrator to add a member repository to a federation. The Documentum Administrator online help contains the instructions.

Removing a member

You cannot remove a member repository while any federation jobs are running. Removing a member from a federation updates the federation object. When a federation job is running, the federation object is locked and cannot be updated.

Removing a member repository also sets the `dm_FederationImport` job in that repository to inactive and hidden. The `a_is_hidden` property is set to `TRUE` for the job.

It is not necessary to make the global objects in the repository local objects. However, if you intend to add the repository to a different federation, we recommend that you make the global objects in the repository local objects first.

Use the federation management facilities of Documentum Administrator to remove a member repository from a federation. The Documentum Administrator online help contains the instructions.

Destroying a federation

You cannot destroy a federation while any federation jobs are running. Destroying a federation updates the federation object. When a federation job is running, the federation object is locked and cannot be updated.

After you remove a federation, the former member repositories still have global users and groups. This operation does not affect the functioning of these repositories. However, we recommend that you make the global users and groups in these repositories local users and groups before adding the repositories to a new federation.

Use the federation management facilities of Documentum Administrator to destroy a federation. The Documentum Administrator online help contains the instructions.

Inactivating a governing repository

Inactivating a governing repository halts all federation-related jobs. To inactivate the governing repository, use the federation management facilities of Documentum Administrator. The Documentum Administrator online help contains the instructions.

User operations

This section contains information for managing the global users within a federation. Use the information in this section only if the global users are common only to the repositories participating in the federation.

A federation's global users are users who are present in all repositories in the federation. Global users are managed through the governing repository. You create a new global user in the governing repository, and the federation jobs automatically propagate the user to all member repositories. Similarly, if you delete a global user from the governing repository, a federation job automatically deletes the user from the member repositories.

Most of a user's properties are global properties. Global property values are the same in each repository in the federation. Global properties must be modified through the governing repository, using Documentum Administrator.

A few properties are local properties. Typically, local properties have different values in each member repository. However, there are four local properties that can behave like global properties. These four properties are `user_state`, `client_capability`, `workflow_disabled`, and `user_privileges`. When you create a global user, you can choose to propagate the values of these properties to all member repositories when the properties are changed in the governing repository.

Note: Sysadmin and Superuser user privileges are never propagated even if you choose to have the `user_privileges` properties managed as a global property. Extended user privileges are not propagated either.

Perform the procedures using Documentum Administrator if your site is not using an LDAP directory server. The procedures are performed on the governing repository, through the federation management facilities of Documentum Administrator.

If your site is using an LDAP directory server to implement global users across all repositories, use the procedures in the LDAP directory server documentation to add or change global user entries in the directory server. Do not use Documentum Administrator.

Creating a user

Use the federation management facilities of Documentum Administrator to add a global user to a governing repository. The Documentum Administrator online help contains the instructions. The online help also has information about user properties.

Use the following guidelines when creating a user:

- The name of the user must be unique among the user and group names in the governing repository.
- A global user's `user_name`, `user_os_name`, `user_address`, and `user_db_name` must be compatible with the server operating system code pages (`server_os_codepage`) of all participating repositories. If the repositories are using different code pages, this means that those properties must contain only ASCII characters.
- If a member repository has a local user with the same name as the global user, the global user overwrites the local user.
- Assigning a global external ACL to the user as the user's default ACL is recommended.

Note: If the ACL assigned to the user does not exist on the member repository, the server uses a system ACL named Global User Default ACL to create a custom ACL for the user. The permissions in Global User Default ACL are `dm_world=3` (Read), `dm_group=5` (Version), and `dm_owner=7` (Delete). Content Server:

- Determines to which group the user belongs.
 - Creates a custom ACL from the Global User Default ACL that references that group.
 - Assigns that ACL to the user.
- If you check a `Send to members?` checkbox, the value of the associated local property is propagated to member repositories, with one exception. The exception occurs for user privileges

and extended user privileges. If the assigned user privilege is Sysadmin or Superuser or any of the extended user privileges, even if the Send to members? checkbox is checked the privilege value is not propagated.

- Documentum client applications use the client capability setting to determine what functionality to make available to the user. Content Server does not use the client capability setting. The client documentation contains more information on the functionality provided with each capability level.

After you create a global user on the governing repository, the federation jobs automatically propagate the new user to the member repositories. The global property values assigned in the governing repository are copied to each member repository. The local properties are assigned default values. The following describes the default values assigned to the local properties in the member repositories.

Property	Default setting
user_db_name	A null string
default_folder	<p>If the member repository is the user's home repository, the default_folder property is set to the default_folder value specified when the user was created in the governing repository.</p> <p>If the repository is not the user's home repository, default_folder is set to /Temp.</p>
user_privileges	<p>The privilege level assigned when the user was created in the governing repository, unless the user was given Sysadmin or Superuser privileges.</p> <p>Superuser and Sysadmin privileges are not propagated automatically, which ensures that member repositories do not acquire unexpected users with these privileges. Only users who are installation owners retain Sysadmin or Superuser privileges when they are propagated to member repositories. All other users with Superuser or Sysadmin privileges in the governing repository are given no privileges in the member repositories (user_privileges=0).</p>
user_state property	The user_state property is set to the value assigned when the user was created.
alias_set_id	The alias_set_id property is not assigned a default value.

Modifying user information

You can perform the following modifications for users:

- Modify global and local property values for a user
- Change the user's default ACL

- Change the user's state
- Change the user's home repository

Modify global users in the governing repository. The Documentum Administrator online help contains the instructions.

Renaming a user

Renaming a user affects more than just the user's `dm_user` object. The server stores a user's name in various properties for different object types. For example, `r_creator_name` and `owner_name` values are user names. User names also appear as values for the `acl_domain` and `authors` properties.

When you run the job that renames a user, it attempts to change all references to that user in the repository. If some of the references are in the properties of locked (checked out) objects, by default the job unlocks the objects (cancels the checkout) and makes the change. If you do not want the job to unlock these objects, be sure to set the Checked Out Objects preference to Ignore before running the job.

You can choose whether to run renaming job immediately or the next time jobs are executed. In a large repository, the rename job can take a long time to complete. It touches many tables and can use a large amount of resources. Running the job during peak usage hours is not recommended.

Rename a user through the governing repository. Use the federation management facility of Documentum Administrator. The Documentum Administrator online help contains the instructions.

Making a local user global

To change a local user to a global user you can use Documentum Administrator or DQL.

Using Documentum Administrator

If the user is local to the governing repository, you can simply change the user to global in the governing repository. The user is then propagated to all members in the next update operation.

If the user is local to a member repository, you must create the user as a new global user in the governing repository for the federation. The new user is automatically propagated to all the member repositories in the next update operation. As part of the process, the user is updated to global in the original member repository also.

Using DQL

To use DQL to change a local user to a global user, use the `UPDATE...OBJECT` statement to set the value of the `globally_managed` property to `TRUE` for the user:

```
UPDATE "dm_user" OBJECT SET "globally_managed"=TRUE WHERE
"user_name"='local_user_name'
```

Making a global user local

To change a global user to a local user, use Documentum Administrator. You make the change in the federation's governing repository. The next federation update operation updates the user information in the member repositories, making the user local.

Deleting a global user

Use Documentum Administrator to delete a global user. Use the federation management facilities to connect to the federation and delete a global user. The Documentum Administrator online help contains the instructions. Deleting a user also deletes all registry objects that reference the user as the subject of an audit or event notification request.

Renaming the user is recommended instead of deleting the user.

Group operations

This section contains the information about procedures for managing global groups within a federation.

Perform the procedures using Documentum Administrator if your site is not using an LDAP directory server. The procedures are performed on the governing repository, through the federation management facilities of Documentum Administrator.

If your site is using an LDAP directory server to implement global groups across all repositories, use the procedures in the LDAP directory server documentation to add or change global group entries in the directory server.

Creating a group

All the properties of a group are global. The values you define for a global group when you create it are propagated to all member repositories. The members of a global group must be global users or other global groups. To create a group, you must have Create Group, Sysadmin, or Superuser privileges.

A global group's `group_address` must be compatible with the server operating system code pages (`server_os_codepage`) of all participating repositories. If the repositories are using different code pages, `group_address` must consist of only ASCII characters.

The name of the group must be unique among the user and group names in the governing repository.

Note: Content Server stores all group names in lowercase.

If a member repository has a local group with the same name as the global group, the global group overwrites the local group.

You can use the federation management facilities in Documentum Administrator to create a global group. The Documentum Administrator online help contains the information.

Alternatively, you can execute DQL CREATE GROUP and UPDATE...OBJECT statements using IDQL or Documentum Administrator. After you create the group, update it to set the `globally_managed` property to TRUE. The syntax is:

```
CREATE GROUP name_of_group WITH MEMBERS list_of_members  
UPDATE "dm_group" OBJECT SET "globally_managed"=TRUE WHERE  
"group_name"='name_of_group'
```

Modifying a group

To modify a group, you must be one of the following:

- The group's owner
- A superuser
- A member of the group that owns the group to be modified
- Identified in the group's `group_admin` property, either as an individual or a member of the group specified in the property

Only a superuser can change the ownership of an existing group. Only a superuser, the owner of a group, or a member of the group that owns the group can change the `group_admin` property of a group.

Use the federation management facilities in Documentum Administrator to modify a global group.

Renaming a group

To rename a global group, use Documentum Administrator. The Documentum Administrator online help contains the information.

Deleting a group

To delete a global group, use Documentum Administrator. You must be a superuser, the group's owner, or a member of the group that owns the group to delete a group. The Documentum Administrator online help contains the information. Deleting a group also deletes all registry objects that reference the group as the subject of an audit or event notification request.

Renaming the group is recommended instead of deleting the group.

Making a global group local

Content Server does not support directly changing a global group to a local group. Instead you create a matching local group and then delete the global group.

Modifying object replication jobs

You can modify a job after it is created. You can change the frequency with which a job is executed, suspend a job, and resume a job.

You can also change the settings for some properties that affect the replicas created by the job. Changing these settings only affects replicas created after the change. Replicas created by previous executions of the job are not affected. For example, if you change the ACL assigned to replicas, subsequent executions assign the new ACL to the replicas created by those executions, but existing replicas created by previous executions retain the original ACL.

If you want to change all replica objects in the target cabinet or folder, delete the replica objects and perform a full-refresh execution of the job.



Caution: Never change the source or destination folder. Running a job stores context information about the source and target cabinets and folders. If you change these parameters, define a new job.

Use the job management facilities in Documentum Administrator to change a job. The Documentum Administrator online help contains the instructions.

Obtaining a list of jobs

Use the job management facilities in Documentum Administrator to view a list of jobs defined for a particular repository. You can view all jobs in the repository or the jobs defined by a category. For example, you can view those jobs that operate on the repository as a whole or only content management jobs. The Documentum Administrator online help contains the instructions on viewing a list of jobs in the repository.

Scheduling federation jobs

Federation jobs manage global users, groups, and external ACLs in the federation. These jobs are executed automatically, in a sequence controlled by the `dm_FederationUpdate` job.

Federation jobs and the methods they invoke are installed as part of the Content Server installation process. The jobs are installed in the inactive state with the `is_hidden` property set to `TRUE`.

When you create a federation, only certain jobs are activated. These jobs are the `dm_FederationUpdate` and `dm_FederationStatus` jobs in the governing repository. If you check **Propagate Global Objects to Member Repositories** when you create the federation, they are activated and made visible automatically. Otherwise, do it manually.

All other federation jobs in the governing repository and member repositories remain inactive and invisible. the `dm_FederationUpdate` job running in the governing repository controls their execution. By default, the `dm_FederationUpdate` job runs once a day, during the night. When it executes, it runs the methods associated with the jobs in the correct sequence for each member repository.

You can change the scheduling for the `dm_FederationUpdate` job on the governing repository. However, we strongly recommend that you do not activate any of the other federation jobs. Doing so causes them to run out of sequence.

Note: To avoid federation job failures, include the `%DM_HOME%\bin` (Windows) or the `$DM_HOME/bin` (on UNIX or Linux) path in the `CLASSPATH` environment variable.

Identifying the federation jobs operator

The federation jobs operator receives email sent by the system reporting the status of the federation jobs. By default, this user is identified in the `operator_name` property of the server's server configuration object. The value in `operator_name` defaults to the repository owner.

If the federation administrator is not the user identified in `operator_name`, reset the `queueperson` argument for the federation jobs to the federation administrator's user name. Use Documentum Administrator to display the Properties screen for each job and reset the `queueperson` argument.

Tracing ACL replication in federation jobs

The `dm_FederationUpdate` job calls the `dm_ACL_Replrepository_name` job to replicate ACLs to member repositories. There is one `dm_ACL_Repl` job for each member repository. By default, the tracing level in the `dm_ACL_Repl` jobs is set to 0. To turn on tracing for the ACL replication jobs, set the `method_trace_level` argument for the jobs.

The tracing information is recorded in the job report generated by the federation update job.

Job reports and log files

Federation jobs and object replication jobs generate reports and log files.

Job reports

A job report summarizes the results of a job in an easily interpreted format. To view reports for federation jobs, you connect to the federation. To view job reports for object replication jobs, connect to the repository that executed the job. Use Documentum Administrator to view job reports.

In the repository, job reports are stored in `/System/Sysadmin/Reports`. Each time a job executes, the job report in this location is versioned.

The reports are also stored in the file system in `%DOCUMENTUM%\dba\log\repository_id\sysadmin` (Windows) or `$DOCUMENTUM/dba/log/repository_id/sysadmin`. In this directory, the report name has the format `job_nameDoc.txt`. The reports in this directory are overwritten each time the job executes.

Federation job reports are also stored in %DOCUMENTUM%\share\temp\ldif\repository_name (Windows) or \$DOCUMENTUM/share/temp/ldif/repository_name for access through Documentum Administrator.

Job log files

Each job execution generates a log file in addition to a report. The log file is typically used for debugging. If you want to know how long a job took to complete or why a job did not run to completion, look in the log file.

For object replication jobs, the log file information includes:

- Whether the job is working in Phase 1, 2, or 3
 - Phase 1 is when the dump file is created
 - Phase 2 is when the file is transferred to the target
 - Phase 3 is when the dump file is loaded into the target repository
- The current job status
- The elapsed time of the job
- The incremental time of the job.
- The incremental time is the length of time since the last status message.
- The elapsed time is the length of time since the job began.

To view federation job log files, you connect to the federation. To view the log files for object replication jobs, connect to the repository that executed the job. Use Documentum Administrator to view job log files.

In the repository, job log files are stored in /Temp/Jobs/job_object_name. The log files in this directory are versioned when the jobs execute.

The log files are also stored in the file system in %DOCUMENTUM%\dba\log\repository_id\sysadmin (\$DOCUMENTUM/dba/log/repository_id/sysadmin). The name of the log file has the format job_nameTrace.txt. The log file in this directory is overwritten each time the job executes.

Federation job log files are also stored in %DOCUMENTUM%\share\temp\ldif\repository_name (\$DOCUMENTUM/share/temp/ldif/repository_name) for access through Documentum Administrator.

The dm_DistOperations job

The dm_DistOperations job performs inter-repository distributed operations. These tasks include:

- Propagating distributed events (dmi_queue_items) across repositories
- Creating checkout references for remote checkout operations
- Refreshing reference links

This job is one of the system administration tools installed with the Content Server installation. The other tools are described in the Documentum Administrator online help.

Like replication jobs, the dm_DistOperations jobs generates reports and log files. The report is saved in the repository in /System/Sysadmin/Reports/DistributedOperations.

The dm_DistOperations job runs continuously in each repository, polling every 5 minutes for operations it performs. The following table describes the job's arguments.

Argument	Value	Description
-process_queued_operations	Boolean	If TRUE, the job processes all dmi_queue_item objects that have remote_pending set to TRUE. The default value is TRUE.
-process_refreshes	Boolean	If TRUE, the job checks for and performs needed reference links refreshes. The default value is TRUE.
-refresh_checks_per_cycle	integer	Defines the maximum number of reference links to check for needed refresh in each job execution. The default value is 100.
superuser and passwords for each repository	string	The job must have a valid superuser name and password for each of the participating repositories. These values are entered as arguments using the format: <i>-server_config_name user_name,password[,domain]</i> server_config_name is the object name of the server configuration object for the repository. The superuser and password must be valid for the repository. They can be the same or different for each repository. domain is required only when unique domain user is enforced.

Monitoring and debugging federation jobs

All federation jobs return one of three values: success, warning, or failure.

A warning indicates that the job encountered a non-fatal error. The job continued to execute but some errors can have occurred during its execution. Warnings typically occur in the following situations:

- The status or copy methods cannot establish a session with a member repository to retrieve the status or copy global objects.
- The status or copy methods cannot validate the `dm_federation` object in a member repository.
- The import method on a member repository experienced difficulties such as:
 - The default folder could not be created
 - The user's default ACL was not found, necessitating the use of the default system ACL

Failures indicate a fatal error that stopped the job's execution. Fatal errors can leave the `dm_federation` object locked in the repository. In such cases, manually check in the federation object.

If a warning or failure occurs:

1. Review the job report.
2. If necessary, review the job log file.

Recovering from replication job failures

Object replication failures generally result from an improperly configured source or target installation or from infrastructure inadequacies, such as insufficient network bandwidth or CPU capacity. If a replication job fails to complete successfully:

- Examine the Properties for the job to determine the last status for the job.
- Review the log file for the job for detailed information on why the job failed.
- If the error message in the trace file suggests that something is configured incorrectly, reconfigure it appropriately and rerun the job.

For example, if the replication job calls for replicas to be stored in a nonexistent file store, create the appropriate file store and rerun the job. The Documentum Administrator online help contains the instructions on running a job outside of its scheduled time.

- Make certain that adequate machine and network resources were available when the job failed.
- If the message in the log file suggests that there were insufficient resources available, rerun the job to see if the same failure occurs.

In general, replication job failures are not destructive. They do not harm existing replicas in the target repository. The next successful execution of the job will clean up after the failure.

Correcting the cause of the failure restores successful replication. However, as part of the process of correcting the problem and rerunning the job to check it, you can reset the job schedule. The Documentum Administrator online help has information about scheduling jobs.

Clearing the replicate_temp_store storage area

If an object replication job is aborted or incompleted, Content Server can leave data in the replicate_temp_store storage area. (This area is used to hold the dump files created by object replication jobs for the duration of the job.) You can remove data left behind under such circumstances.

The directory location of the replicate_temp_store storage area for a particular repository is %DOCUMENTUM%\data\replicate_temp_store\repository_id (\$DOCUMENTUM/data/replicate_temp_store/repository_id), where repository_id is the hex value.

Handling replicas

This section contains procedures and information about management of replicas in a repository.

Defining a binding label for reference links

The binding between a mirror object and its source object is typically set by default when the user executes the operation that creates the reference link. For example, suppose a user, who is logged in to repository A, checks out the approved version of DocumentX, which resides in repository B. The server creates a reference link in repository A to the approved version of DocumentX.

If the user's operation does not identify a specific version of the object, the server binds the CURRENT version to the reference link by default.

You can override the binding specified at runtime by setting the ref_binding_label property in the session configuration object. For example, suppose ref_binding_label is set to CURRENT before the user checks out the approved version of DocumentX. Even though the user requests the approved version, the server binds the CURRENT version to the reference link. When the user opens the document, the server opens the CURRENT version. The setting in ref_binding_label overrides the binding label identified by the user.

If the version defined in ref_binding_label does not exist, the server returns an error.

To return to the default behavior, set the ref_binding_label property to a blank string.

Determining whether an object is a replica

Replica objects are identified in the user interface by a replica icon. Programmatically, you can identify a replica using the computed property _isreplica. This property is set to T for replicas. You can query this property using the Get method or the DQL SELECT statement. For DQL, specify the property name without the underscore (isreplica).

Determining a replica's source

To determine the source of a replica, examine its reference object or query the computed property `_masterdocbase` for the replica. This property returns an object's source repository.

Federation infrastructure

The table in this appendix describes the jobs, methods, files, and objects that support federations.

Object name	Type	Description
dm_FederationUpdate	job/method	Executes on the governing repository to run all the other methods in sequence, to push changes to users, groups, and ACLs to the member repositories. The job and its associated method have the same name.
dm_FederationStatus	job/method	Polls all the member repositories to determine the status of the dm_FederationImport jobs on the member. The job and its associated method have the same name.
dm_ldif_status	method	Called by dm_FederationStatus to poll the member repositories.
dm_FederationExport	job/method	Exports the user and group information from the governing repository to an LDIF file. The job and its associated method have the same name.
dm_ldif_export	method	Called by dm_FederationExport to generate an LDIF export file.
dm_FederationCopy	job/method	Transfers LDIF files to member repositories. The job and its associated method have the same name.

Object name	Type	Description
dm_ldif_copy	method	Called by dm_FederationCopy to transfer the LDIF files from the governing repository to member repositories.
dm_FederationImport	job/method	Imports an LDIF file into member repositories.
dm_ldif_import	method	Called by dm_FederationImport to import the users and groups from the LDIF file into a member repository.
dm_ACLReplication	job/method	A staging job that sets external ACLs for replication.
dm_ACLRepl_repository	job	Replicates external ACLs to member repositories. There is one job for each member repository. <i>repository</i> is the first 19 bytes of the repository name.
ldif	directory	Contains the LDIF files generated by dm_FederationExport. The directory is found in %DOCUMENTUM%\share\temp (\$DOCUMENTUM/share/temp).
dm_federation_log	registered table	Tracks certain changes to users and groups. This table is the change log.

Documentum Administrator

This chapter describes how to deploy the Documentum Administrator application. The contents are intended for administrators who are deploying Documentum Administrator.

Documentum Administrator is a Content Server and repository administration tool. Documentum Administrator runs on an application server host. The intended audience are expected to be familiar with the Windows, UNIX, or Linux operating systems and are able to install and configure a J2EE application server.

EMC Documentum Content Server Administration and Configuration Guide and *Documentum Administrator online help* contain information on how to use Documentum Administrator to administer and configure Content Server and Documentum repositories.

Planning for deployment

Required and optional supporting software

Before deploying Documentum Administrator (DA), the following components must be installed:

- Content Server and its associated database
- Content Server global repository
- Connection broker
- J2EE application server or servlet container

Typical configuration

When deployed on a single application server, Documentum Administrator requires the following network components:

- Application server host on which to deploy Documentum Administrator
- Separate Content Server host with a repository and one or more Content Servers
- Global registry repository
- Client hosts that run a supported web browser

Documentum Administrator can be deployed in supported clustered environments.



Caution: In production systems, for security and performance reasons, do not install the Content Server and Documentum Administrator on the same host. Also, do not deploy web applications to the internal application server embedded in the Content Server. Non-xCP Documentum applications (such as Documentum Administrator, Webtop) cannot be deployed on the application server instance where xCP runtime is hosted because of conflicting `dfc.jar` instances on the classpath. Do not deploy Documentum Administrator on the same application server where xCP is deployed.

Application server host requirements

The application server host used for Documentum Administrator requires the following:

- Directory name restriction

Java does not allow directories containing the following characters, which must not appear in the directory names or paths of Documentum applications:

! \ / : * ? " < > |

- Content transfer directory permissions

The content transfer directory on the application server host is used to store files temporarily when they are transferred between the repository and the client machine. The default content transfer directory is specified in the `<DA_WAR_DEPLOYMENT>/wdk/app.xml` file as the value of `<server>,<contentlocation>`. The application server instance owner must have write permissions on this temporary content transfer location.

Some application servers require policies that grant permissions to write to these directories. Refer to deployment information for your application server to see Documentum policy settings.

- DNS resolution

The Domain Name Server (DNS) must be configured properly to resolve IP addresses based on the URL used to access the server.

Customizing Documentum Administrator

Customization of Documentum Administrator is not supported.

Preparing the client hosts

Ensuring a certified JVM on browser clients

Browser client hosts require a certified version of the Java virtual machine (JVM or VM) to initiate content transfer in Documentum Administrator.

For UCF content transfer, UCF downloads a lightweight applet to the browser when the client makes the first content transfer or preferences request. If the JVM required for UCF is not present on a Windows client, UCF uploads a private JVM that does not affect the browser JVM.

Enabling HTTP content transfer in Internet Explorer

Internet Explorer has a default security setting that prevents the display of the file download dialog for checkout, view, and edit operations in HTTP mode. To resolve this, add the Documentum Administrator URL to the list of trusted sites in the browser.

If the browser security settings are disabled for **Automatic prompting for file downloads** and **File download**, nothing happens when a user exports as CSV. These settings are disabled by default in Internet Explorer. The user must enable them and then restart Internet Explorer.

1. In Internet Explorer, navigate to **Tools > Internet Options** and click the **Security** tab.
2. Select **Trusted sites** and click **Custom level**.
3. Scroll to the **Downloads** section and enable **Automatic prompting for file downloads** and **File download**.
Click **OK** twice to save the settings.
4. Close all browser windows and restart the browser.

Preparing the application server host

Application servers

Before deploying Documentum Administrator, ensure that your J2EE application server or servlet container is a supported version that serves sample JavaServer Pages successfully. Your selected application server and optional external web server must be certified for Documentum Administrator.

EMC does not provide support for installing or running application servers. The documentation for each application server contains instructions on how to install, stop, start, and run the application server. Contact the application server vendor for technical support.

Setting the Java memory allocation

Java memory allocation settings affect the application server performance. We recommend using the following settings:

- Minimum memory allocation

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

```
-Xms1024m -Xmx1024m
```

- MaxPermSize

Application servers can slow down, throw exceptions, or crash with an application that has many JavaServer Pages. Set the MaxPermSize parameter to 128 or higher to avoid these problems.

- Session caching

Document caching can consume at least 80 MB of memory. User session caching can consume approximately 2.5 MB to 3 MB per user. Consequently, 50 connected users can consume over 200 MB of VM memory on the application server. Increase the values to meet the demands of the expected user load.

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

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

The first parameter on the command line must be **-server**.

Performance improves because the Java client VM is not suitable for long running server jobs.

The default Java garbage collector cannot clean up the heap quickly enough, especially when the application server machine runs on multiple CPUs.

The *Java documentation* contains more information on these settings. More information on application server performance tuning and benchmarking for Documentum products is available from your EMC Documentum SE or EMC Documentum Consulting.

Turning off failover

If your application server and environment combination does not support failover, you can disable failover in `custom\app.xml` by adding the following element:

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

If you do not turn off failover, you see failover validation messages in the application server log, but these validations do not interfere with operations. Do not use the application in a failover environment.

Preparing environment variables for non-default DFC locations

The DFC environment variable `dfc.data.dir` specifies the base location for content transfer on the application server host. This location is specified as the value of the key `dfc.data.dir` in the `dfc.properties` file located within the application WAR file in `WEB-INF/classes`. If this variable is not set in the environment for the application server, the default location is the Documentum subdirectory of the current working directory. (The current working directory contains the application server executable.) For example, in Apache Tomcat the location is `<CATALINA_HOME>/bin`. On Oracle WebLogic, it is `<BEA_HOME>/domains/wl_server/documentum`.

By default, the checkout and export directories are subdirectories of the `dfc.data.dir` directory, and the user directory is the same as `dfc.data.dir`. If you wish to use non-default locations for these directories, create environment variables for `dfc.checkout.dir`, `dfc.export.dir`, and `dfc.user.dir`, respectively. The default value of `dfc.registry.mode`, which corresponds to the key `dfc.registry.mode` in the `dfc.properties` file, is *file*. By default, the full path to this file is `dfc.user.dir/documentum.ini`. For a non-default file name or location, specify it as the value of the environment variable `dfc.registry.file`.

Configuring Apache Tomcat

In Apache Tomcat, the `HttpOnly` property of cookies is enabled by default and causes the `jsessionid` cookie to be unavailable to the client side script and applets. Hence, perform the following and then restart the web application server:

- Modify the `useHttpOnly` tag in the `context.xml` to **false**.
- Add the following line in the `catalina.properties` file located at `<APACHE_TOMCAT_HOME>/conf`:

```
org.apache.jasper.compiler.Parser.STRICT_WHITESPACE=false
jnlp.com.rsa.cryptoj.fips140loader=true
```
- Disable tag reuse in Apache Tomcat in the `web.xml` file of the `/conf` directory. Find the JSP servlet entry in the `web.xml` file. Add the `enablePooling` initialization parameter, disable pooling, and then restart the web application server:

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>enablePooling</param-name>
    <param-value>false</param-value>
  </init-param>
  <init-param>
  </init-param>
</servlet>
```

When deploying Documentum Administrator on Tomcat 8, compression must be set to the compression mode of the web application server. For better performance on Tomcat 8.x, do the following:

- Enable web application server compression: Navigate to the `<Tomcat Home>/conf` directory, open `server.xml`, search for Connector tag, and then append the following entry to the Connector tag:

```
compression="on"
compressionMinSize="2048"
compressableMimeType="text/html,text/xml,application/xml,text/plain,text/css,text/
javascript,text/json,application/x-javascript,application/
javascript,application/json"
useSendfile="false"
```

- Disable the WDK compression filter: Open the custom/app.xml file, search for the <compression_filter_enabled> tag, set it to false, and then restart the web application server.

Configuring JBoss EAP

1. If available, delete the dfc.keystore and wdk.keystore files in <JBoss Home>\bin (Windows) and <JBoss Home>/bin (Linux).
2. Move the keystore files from <WebApp Root>\WEB-INF\classes (Windows) and <WebApp Root>/WEB-INF/classes (Linux) to the bin folder of the <JBoss Home> directory.
3. Copy the contents of the classes folder from <WebApp Root>\WEB-INF\classes (Windows) and <WebApp Root>/WEB-INF/classes (Linux) to a temporary location (for example, Temp-Loc).

Execute the following command at Temp-Loc to create a web-inf-classes jar file:

```
jar -cvf web-inf-classes.jar *
```

4. Copy the web-inf-classes.jar file to <WebApp Root>\WEB-INF\lib (Windows) and <WebApp Root>/WEB-INF/lib (Linux).
5. Delete the classes folder from <WebApp Root>\WEB-INF (Windows) and <WebApp Root>/WEB-INF (Linux).
6. Add the configuration entry (in bold) to the subsystem tag in standalone.xml in <JBoss Home>\standalone\configuration (Windows) and <JBoss Home>/standalone/configuration (Linux) to disable tag pooling:

```
<subsystem xmlns="urn:jboss:domain:web:2.1"
default-virtual-server="default-host" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<virtual-server name="default-host" enable-welcome-root="true">
<alias name="localhost"/>
<alias name="example.com"/>
</virtual-server>
<configuration>
<jsp-configuration tag-pooling="false"/>
</configuration>
</subsystem>
```
7. Configure the binding address by replacing 127.0.0.1 with the application server host IP address in <wsdl-host> and <interfaces> tags in standalone.xml.

8. Execute the following command at <WebApp Root> to repack the DA WAR file:

```
jar -cvf da.war *
```

When deploying Documentum Administrator in JBoss EAP, add

```
<path name="com/sun/jndi/url/rmi"/>
to <jboss-deployment-folder>\modules\system\layers\base\sun\jdk\main\module
.xml under the <paths> tag.
```

Configuring VMware vFabric tc Server

In VMware vFabric tc Server, the `HttpOnly` property of cookies is enabled by default and causes the `jsessionid` cookie to be unavailable to the client side script and applets. To fix this issue, perform the following and then restart the web application server:

- Modify the `useHttpOnly` tag in the `context.xml` to **false**.
- Add the following line in the `catalina.properties` file located at `<VMware_vFabric_tc_Server_HOME>\conf`:
`org.apache.jasper.compiler.Parser.STRICT_WHITESPACE=false`

Preparing IBM WebSphere

Running Documentum Administrator on an IBM WebSphere application server requires the following:

- Preparing the application server to support failover in a cluster
- Applying policies for Java 2 security
- Supporting non-default content transfer locations

Disabling HttpOnly Property

Deselect **Set session cookies to HTTPOnly to help prevent cross-site scripting attacks** from the location `Application servers>server1>Session management>Cookies`.

Note: If there are multiple applications deployed in the same application server and if you require to set the flag `HttpOnly` just for WDK application, then perform the following steps:

1. Deselect **Set session cookies to HTTPOnly to help prevent cross-site scripting attacks** from `All Applications>da>Session management>Cookies`.
2. Check Override the session management from All Application `>da>session management`.

Supporting failover in a cluster

Failover in a clustered environment requires that you set the `NoAffinitySwitchBack` custom property to `true` in the WAS cluster. The *IBM WebSphere documentation* contains more information on this setting.

Applying policies for IBM WebSphere security

If IBM WebSphere global security is enabled for the application server, by default it enables Java 2 security. Java 2 security requires security policies. Apply the policies in the Documentum

files `app.policy`, `library.policy`, and `was.policy` available in the compressed archive `PolicyFiles.zip`. These files contain a list of security policies (permissions) that will be enabled when WebSphere is started. Add these policies to WebSphere policy files. The *IBM Websphere documentation* contains more information.

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

- `dfc.data.dir`

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

- `webtop.content.xfer`

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

To add policies for non-default content transfer locations, add the following lines to `library.policy`. For each policy that you add, set up an environment variable that specifies the non-default location.

Policy for Documentum Administrator —

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

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

```
permission java.io.FilePermission "${dfc.user}${/}-", "read, write, delete";
permission java.io.FilePermission "${dfc.user}", "read, write, delete";
```

Policy for checkout and export directories (non-default location) — These environment variables must specify the same location as the value of the `dfc.properties` keys `dfc.checkout.dir` and `dfc.export.dir`. The default locations for these directories are checkout and export subdirectories of `dfc.data.dir`.

```
permission java.io.FilePermission "${dfc.checkout}${/}-", "read, write, delete";
permission java.io.FilePermission "${dfc.checkout}", "read, write, delete";
permission java.io.FilePermission "${dfc.export}${/}-", "read, write, delete";
permission java.io.FilePermission "${dfc.export}", "read, write, delete";
```

Policy for DFC registry file (non-default location) — The value of the `dfc.registry` environment variable must match the location specified in the `dfc.properties` file for the key `dfc.registry.file`.

```
permission java.io.FilePermission "${dfc.registry}${/}-", "read, write, delete";
permission java.io.FilePermission "${dfc.registry}", "read, write, delete";
```

Policy for Webtop temporary content transfer directory (non-default location) —

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

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


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

Policy for documentum applications directory (non-default location) — The default location is `dfc.data.dir`.

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

Policy for DFC class cache directory (non-default location) — The default location is `dfc.data.dir/cache`.

```
permission java.io.FilePermission "${dfc.cache.dir}${/}-", "read, write, delete";
permission java.io.FilePermission "${dfc.cache.dir}", "read, write, delete";
```

Policy for Content Intelligence Services —

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

Preparing Oracle WebLogic

If you are deploying in a Oracle WebLogic Managed Server environment and use UCF to perform large content operations, set the `WLIOTimeoutSecs` parameter for the web server plug-in to a large value. UCF requires a sticky session for a single operation. The *Oracle WebLogic documentation* on Web Server Plug-ins parameters contains additional details.

When deploying Documentum Administrator along with D2 application on the same application server, add the following lines to the `weblogic.xml` present in `<DA>\WEB-INF` folder.

```
<session-descriptor>
<cookie-path>/DA</cookie-path>
</session-descriptor>
```

Disabling HttpOnly property

Modify the `cookie-http-only` tag in the `WebLogic.xml` to **false** and then restart the web application server.

Preparing the application server for Java 2 security

If you plan to use Java 2 security for securing access to available system resources in your Documentum Administrator installation, then use the java policy configuration file that is bundled with your application server. To help you update the java policy configuration file of the application server, an example policy template file is included in the Documentum Administrator installation (`Webtop.example.java.policy` file). The file specifies the permissions required to access the Documentum Administrator classes. The `Webtop.example.java.policy` file is included in the `da.war` file, and gets extracted into the `<da_app_root>` folder.



Caution: Do not omit any permission specified in the `Webtop.example.java.policy` file while incorporating the permissions in the application server java policy configuration file. Otherwise, Documentum Administrator might fail to start or some features might fail to work.

1. Navigate to `<da_app_root>\Webtop.example.java.policy` and identify the permissions that must be incorporated into the application server security policy file.
2. Navigate to the policy file of your application server.
Based on the syntax and locations specified in the application server documentation, add or update the permissions (identified in the `Webtop.example.java.policy` file) in the policy file of the application server.
3. Configure your application server to pick the security policy files.

Preparing to use an external web server

External web servers are sometimes used as a front end to the application server. If the external server does not support HTTP 1.1 chunked encoding, configure UCF to use an alternative chunked encoding.

If you are deploying in a manager server or network deployment environment, the external web server must provide session affinity support.

Deploying Documentum Administrator

Prerequisites

Ensure you follow described in the pre-installation checklist.

Use the following checklist to verify that you have performed all required tasks when you install or upgrade DA.

Requirement	For more information
Review the release notes for the release you are installing or to which you are upgrading.	<i>EMC Documentum Platform and Platform Extensions Release Notes</i>
Validate your hardware configuration.	
Validate your application server and clients operating systems.	
Create required operating system accounts.	Network administrators
Verify that the application server instance owner has write permissions on the temporary content transfer directories.	Network administrators
Determine the repositories to which end users connect.	Network administrators

Requirement	For more information
Determine the connection brokers to which the repositories project.	Network administrators
Determine which repository on the network is the global registry repository, and obtain the global registry user name and password.	Network administrators
Determine which repositories are used to store presets and user preferences.	Network administrators
Determine whether language packs are required.	<i>EMC Documentum Platform and Platform Extensions Release Notes</i>
Prepare the application server host and application server software according to the vendor's requirements.	Specific requirements should be met for the application server host
Disable the IP Helper service from the Windows Services console and restart the machine. This method disables the Teredo Tunneling Pseudo-Interface.	<i>EMC Documentum Platform and Platform Extensions Release Notes</i>

Deploying the WAR file

Perform the following to deploy Documentum Administrator:

1. Unpack the WAR file and modify the `dfc.properties` file

Before Documentum Administrator can connect to repositories, provide connection broker and global registry values in the `dfc.properties` file.

Documentum Administrator requires a Content Server version 6 or later global registry. The global registry is a central repository that serves several purposes:

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

EMC Documentum Platform and Platform Extensions Installation Guide contains information about enabling a repository as a global registry.

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

Perform the following to locate the `dfc.properties` file values:

1. On the global registry repository host, locate the Content Server installation directory. On Windows hosts, the default installation directory is `C:\Documentum`. On UNIX hosts, the `$DOCUMENTUM` environment variable specifies this directory.

2. Open `config\dfc.properties`.
3. Copy the following keys and their values from the file:

```
dfc.docbroker.host[0]=address
dfc.docbroker.port[0]=port_number
dfc.globalregistry.repository=repository_name
dfc.globalregistry.username=username
dfc.globalregistry.password=encrypted_password
dfc.crypto.repository=repository_name
dfc.session.secure_connect_default=try_secure_first
```

To configure connections in `dfc.properties` file before deployment:

1. Unpack the application WAR file.
2. Open `WEB-INF/classes/dfc.properties`.
3. Add the fully qualified host name for the connection broker to the following key. You can increment the index number within brackets to add backup hosts.

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

4. To use a port for the connection broker other than the default of 1489, add a port key to the `dfc.properties` file:

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

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

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

6. Add the user name of the `dm_bof_registry` user to the following key:

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

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

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

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

You can either copy the username and encrypted password from the `dfc.properties` file on the global registry Content Server host or you can select another global registry user and encrypt the password using the following command:

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

Note: The directory containing the `javaw.exe` file must be on the system path.

8. If the Content Server, connection broker, and the repository are configured in the non-anonymous SSL mode then provide these parameters in the `dfc.properties` file:

- a. Add the secure connection mode and set it to secure first.

```
dfc.session.secure_connect_default = try_secure_first
```

- b. Add the trust store path.

```
dfc.security.ssl.truststore=<dfc truststore path>
```

- c. Add the trust store password.

```
dfc.security.ssl.truststore_password=<password>
```

- d. Specify whether to use the existing trust store.

```
dfc.security.ssl.use_existing_truststore=<false/true>
```

- e. Specify the crypto repository to connect.

```
dfc.crypto.repository=repository_name
```

9. Save the `dfc.properties` file.

Note: If you create a WAR file from this application directory, ensure that any paths that you specify in the `dfc.properties` file are valid directories on the application server. Also ensure that the application server instance owner has `write` permission on the specified directories.

10. Enable and configure the optional presets and preferences repositories in the `dfc.properties` file.

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

Add your preferences repository settings to `app.xml` in the `/custom` directory of the application. Copy the entire `<preferencesrepository>` element from `/custom/app.xml` into `/custom/app.xml` and then specify your repository.

Table 14. Preferences configuration elements

Element	Description
<code><preferencesrepository></code>	Contains a <code><repository></code> element. If this element is not present, user preferences are stored in the global repository, which can slow down performance.
<code><repository_path></code>	Specifies the path within the preference repository in which to store preferences. If the path does not exist at application startup, then it is created.
<code><repository></code>	Specifies the repository in which to store preferences, preferably not the global repository.

To enable users to create presets using the presets editor, assign those users the `dmc_wdk_presets_coordinator` role.

To configure the password in presets and preferences repositories, perform the following steps:

1. Login to IAPI as an administrator to change the default passwords of `dmc_wdk_presets_owner` and `dmc_wdk_preferences_owner` users in Content Server.

- To change the password for the `dmc_wdk_presets_owner` user, run the following command:

```
retrieve,c,dm_user where user_name='dmc_wdk_presets_owner';
set,c,l,user_password
<enter new password>
save,c,l
```

- To change the password for the `dmc_wdk_preferences_owner` user, run the following command:

```
retrieve,c,dm_user where user_name='dmc_wdk_preferences_owner';
set,c,l,user_password
```

```
<enter new password>
save,c,l
```

2. Encrypt the passwords in Documentum Administrator using TrustedAuthenticatorTool located at WEB-INF/classes.

On Windows — Run the following command:

```
java TrustedAuthenticatorTool <password>.
```

The utility sends the encrypted password to the standard output. For example,

```
C:\DA\WEB-INF\classes>java -cp ../lib/dfc.jar;../lib/commons-io-1.2.jar;
../lib/certj.jar;../lib/jsafeFIPS.jar TrustedAuthenticatorTool trusted
Encrypted: [5P54fOKuCKM=], Decrypted: [trusted]
```

On Linux — Perform the following steps:

- a. Navigate to the WEB-INF/classes folder.

- b. Set the classpath for the referenced jars:

```
set JAR_PATH=../lib/dfc.jar:../lib/commons-io-1.2.jar:
../lib/certj.jar:../lib/jsafeFIPS.jar:
```

- c. Execute the Java command to generate the encrypted password:

```
java -cp %JAR_PATH% TrustedAuthenticatorTool trusted
Encrypted: [5P54fOKuCKM=], Decrypted: [trusted]
```

3. Update the encrypted passwords in Documentum Administrator app.xml. Search for <presets> and update the <password> attribute with the encrypted password. For example,

```
<presets>
...
<password>5P54fOKuCKM=</password>
...
</presets>
```

Search for <preferencesrepository> and update the <password> attribute with the encrypted password. For example:

```
<preferencesrepository>
...
<password>5P54fOKuCKM=</password>
...
</preferencesrepository>
```

4. (Optional) Add language packs to and configure them in the DA WAR file.

- a. Unpack the language pack zip file into the root DA WAR directory.

- b. Add the required locale under <supported_locales> in da/custom/app.xml.

For example, for the Japanese language pack, add <locale>ja_JP</locale> to da/custom/app.xml as follows:

```
<supported_locales>
  <locale>en_US</locale>
  <locale>ja_JP</locale>
</supported_locales>
```

5. Rearchive the WAR file.
6. Deploy the WAR file according to the deployment instructions in your application server documentation.

7. (Optional) If you have installed the Japanese language pack and the repository is on a non-Japanese operating system, then you must populate the data dictionary with the Japanese data dictionary files by running `dd_populate.ebs` on a Japanese operating system. *EMC Documentum Content Server Administration and Configuration Guide* contains more information about populating the data dictionary in a repository from a non-English host.

Note: If you have created a repository on a Japanese operating system, then the data dictionary is automatically populated with the Japanese data dictionary files.

Enabling DFC memory optimization

DFC diagnostics are enabled by default. To free up memory resources, disable the `dfc.diagnostics.resources.enable` parameter in the `dfc.properties` file. Add the following line to your `dfc.properties` file:

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

Configuring UCF

EMC Documentum Web Development Kit Development Guide contains more information.

Forcing UCF to install a configured JRE

If Documentum Administrator uses UCF content transfer, it is mandatory that the browser has a JRE installed. By default, the UCF installer uses the JRE that is installed in the browser if its version is the same as or later than the version of JRE in the UCF installer. A later version of JRE sometimes introduces problems in an application.

If you do not want to allow multiple JRE versions, you can configure the UCF installer to use or install only the version that is configured in the installer configuration file. If that version is already installed, the UCF installer uses it. If it is not present, the UCF installer installs and uses the configured version. You must add an `enforceJreInstallation` attribute to the runtime java element in the file `ucf.installer.config.xml` to use the configured JRE version. This file is located in your web application directory, `wdk/contentXfer`. Change the runtime java element by adding the `enforceJreInstallation` attribute as follows:

```
platform os="windows" arch="x86">
<runtime type="java" version="1.7.0_72" href="win-jre1.7.0_72.zip"
exePath="jre1.7.0_72\bin\java.exe" enforceJreInstallation="true">
```

If users have already installed UCF, force an update of the UCF configuration every time you change the UCF configuration on the application server. Ensure that you append a new character to the app element's version attribute to force the update. In the following example, `7.2.223` is changed:

```
<app id="shared" version="7.2.223" compatibilityVersion="7.2"/>
```

Enabling retention of folder structure and objects on export

To enable retaining the same folder structure (as the one in the repository) and the contained objects on the local file system when the parent folder is exported, add the following element to your `app.xml` in the custom directory:

```
<deepexport>
  <enabled>true</enabled>
</deepexport>
```

The default is false.

Enabling external searches

To allow users to search external sources, an administrator must configure a connection to a Federated Search server.

Configuring the connection to the search server

The following procedure describes how to enable the Federated Search server to query external sources.

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

Configuring the connection to the backup search server

You can set a backup server in case the primary Federated Search server is unreachable. If a DFC-application cannot connect to the primary Federated Search server to query external sources, the backup server is contacted. You can define the time period after which the application tries to

connect again to the primary server. To define the backup server, specify the RMI host and port in the `dfc.properties` file:

- `dfc.search.ecis.backup.host`: Host of the backup Federated Search server. Default value is: `localhost`.
- `dfc.search.ecis.backup.port`: Port of the backup Federated Search server. Default value is: `3005`.
- `dfc.search.ecis.retry.period`: Waiting period before retrying to connect to the primary Federated Search server. This time is in milliseconds. Default value is: `300000`.

Requirement for full-text indexing

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

Resource Management availability

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

Enable presets for Administrator Access and Resource Management

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

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

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

Modal popup

When you invoke a component that has been configured for modal popup, the user interface for the component is displayed in a modal popup window. This modal popup window is placed on top of

the current window. The title of the modal popup window shows the title of the component page followed by — **Webpage Dialog**. You can resize the modal popup window but cannot access the parent window until you dismiss the popup window (also known as child window). When you try to close a modal popup window by clicking the [X] button on the window, the framework treats it as a canceling an action.

When you invoke another component that is configured for modal popup from the child window, another modal popup window is placed on top of the child window to show the component user interface. With stacked modal windows, you cannot access a parent window until you dismiss the child window.

Modal popup is only supported in Internet Explorer, but in the 508 accessibility mode.

Configuring the modal popup

You can configure a nested component to display in a modal popup. If a component is tied to an action, you can modify the action definition by adding the `<invocation>` element.

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

This configuration is added to the action definition because the modal popup behavior is tied to how a component is invoked. The idea is to have the modal popup configuration in the action definition. In the invocation element, you can specify the size of the modal popup and whether the framework must refresh the parent window when the child window is closed. All action controls read the configuration. If the configuration indicates that the component tied to this action displays in a modal popup, it opens a modal popup window and submits the request to the component during action invocation. The response is displayed in the modal popup window.

Deploying and configuring Documentum Administrator on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.
3. Provide all the required details in the `statelessda.conf` file. Read the description of every field and provide valid values for each parameter.
4. Run the `statelessda_config.sh` script.

Once the Documentum Administrator image is created, a confirmation message appears.

Also, once the application server is up and running, Documentum Administrator is accessible.

5. To verify the installation, access the URL of Documentum Administrator inside the container.

Note: If you encounter the `STRICT_WHITESPACE` error while accessing Documentum Administrator, access the docker container and perform the following:

1. Run the following commands:

```
$ docker exec -it <containerID> bash
> ./opt/tomcat/bin/shutdown.sh &
> vi /opt/tomcat/conf/catalina.properties
```

2. Add the following line:

```
org.apache.jasper.compiler.Parser.STRICT_WHITESPACE=false
```

3. Save the file.

4. Run the following command:

```
> ./opt/tomcat/bin/startup.sh &
```

Post-deployment tasks

Configuring IBM WebSphere

1. Navigate to **Application Servers > Server1 > Web container > Custom Properties in Admin console** and set the `com.ibm.ws.webcontainer.invokefilterscompatibilitycustom` property to True.
2. Add the `dfc.diagnostics.resources.enable=false` parameter in the `dfc.properties` file of Documentum Administrator.
3. Change the classloader setting for the WDK-based application module in IBM WebSphere in the **Manage Modules** section of the administration console.
 - a. Select the WAR file.
 - b. For **Classloader order**, choose **Classes loaded with local class loader first (parent last)**.
 - c. Click **Save**.
4. Restart the application server.

Configuring Oracle WebLogic class loading behavior

Oracle WebLogic classloader precedence can cause SSL validation to fail. Configure the Oracle WebLogic class loading behavior to load the application level classes first, instead of the Oracle classes.

1. Navigate to the `.\WEB-INF\classes` folder and open the `weblogic.xml` file.
2. Modify the file as follows:

```
<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//
DTD Web Application 8.1//EN" "http://www.bea.com/servers/wls810/
```

```
dtd/weblogic810-web-jar.dtd">
<weblogic-web-app>
<description>Weblogic Webapp</description>
<container-descriptor>
<prefer-web-inf-classes>true</prefer-web-inf-classes>
</container-descriptor>
</weblogic-web-app>
```

3. Save your changes.

Configuring UCF on Oracle WebLogic Server

Oracle WebLogic Server 11g and later requires a modification in the `weblogic.xml` file to configure UCF clients. Without the modification, the Content Server throws an exception when users attempt to view the server log file in Documentum Administrator.

1. Navigate to the `.\WEB-INF\classes` folder and open the `weblogic.xml` file.
2. Add the following lines:

```
<session-descriptor>
<cookie-http-only>false</cookie-http-only>
</session-descriptor>
```

3. Save your changes.

Configuring single sign-on for security servers

Content Server supports authentication plug-ins, SSO using RSA Access Manager (formerly known as ClearTrust), or CA SiteMinder.

RSA Access Manager users must have the same login names as the Content Server repository. User names are case sensitive for the Content Server, so Access Manager user names must be at least 8 characters in length and have the same case as the repository login. Errors in authentication are logged in the `/Documentum/dba/log/dm_rsa.log` file.

For CA SiteMinder, set up a SiteMinder realm to perform authentication for Documentum Administrator. The `dm_netegrity` plug-in installed in the Content Server decodes the `SMSESSION` token sent from Documentum Administrator for authentication. The plug-in contacts the CA server to verify that the token is valid. Errors in authentication are logged in the `/Documentum/dba/log/dm_netegrity.log` file.

The *RSA or CA security server or Content Server documentation* contains more information.

To enable single sign-on (SSO):

1. Configure the RSA Access Manager or CA SiteMinder security server to authenticate repository users.
2. Configure the web application server to use an external HTTP Server supported by the security server.
3. Configure the Content Server plug-in.

4. Configure Documentum Administrator in the `app.xml` file.
5. RSA only: Create a directory named `rsaConfig` under the Documentum Administrator root directory. Copy two files: `aserver.conf` from the Access Manager server and `webagent.conf` from the RSA web agent. Paste them into the `rsaConfig` directory.
If you change the original files, copy them to your Documentum Administrator `rsaConfig` directory.
6. Locate the file `AuthenticationScheme.properties` in `WEB-INF/classes/com/documentum/web/formext/session`. The SSO authentication scheme classes. Modify the properties file to make your preferred SSO authentication scheme (`SSOAuthenticationScheme` or `RSASSOAuthenticalScheme`) first in the list of authentications that are attempted during login.
If the repository login scheme is listed before the SSO scheme, the user is presented with a login screen instead of single sign-on.
7. Restart the application server.

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

The WDK SSO Authentication Scheme for CA SiteMinder needs three pieces of information to authenticate an HTTP session against a repository:

- Name of the authentication plug-in that is used in the Content Server.
 - Name of the ticket to retrieve from a vendor-specific cookie.
 - User name, which is retrieved from a vendor-specific HTTP requests header or remote user.
1. Open the `app.xml` file in your `applications/custom` directory.
 2. Copy from `app.xml` the `<authentication>` element and its entire contents, and paste into your custom `app.xml` file.
 3. Update the `<sso_config>` element under the existing `<authentication>` element as shown in the following example:

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

Note: This example is for RSA.

The following table describes the authentication elements.

Table 15. Authentication elements (<authentication>)

Element	Description
<docbase>	Specifies default repository name. When SSO authentication is enabled but a repository name is not explicitly spelled out by the user nor defined in this element, the <code>sso_login</code> component is called. In this case the component prompts the user for the repository name.
<domain>	Specifies Windows network domain name.
<service_class>	Specifies fully qualified name of class that provides authentication service. This class can perform pre- or post-processing of authentication.
<sso_config>	Contains SSO authentication configuration elements.
<sso_config>	Specifies name of the Content Server authentication plug-in (not the authentication scheme name). Valid values:
<ecs_plug_in>	<ul style="list-style-type: none"> • RSA: <code>dm_rsa</code> • CA: <code>dm_netegrity</code>
<sso_config>	Specifies name of vendor-specific cookie that holds the sign-on ticket. Valid values:
<ticket_cookie>	<ul style="list-style-type: none"> • RSA: <code>CTSESSION</code> • CA: <code>SMSESSION</code>
<sso_config>	Specifies name of vendor-specific header that holds the username. Valid values:
<user_header>	<ul style="list-style-type: none"> • RSA: <code>HTTP_CT_REMOTE_USER</code>. • CA: The <code>user_header</code> value is dependent on the settings in the <code>webagent</code> configuration object in the policy server. The default is either <code>SMUSER</code> or <code>SM_USER</code>, depending on whether the <code>LegacyVariable</code> flag is set to true or false. If true, use <code>SM_USER</code>. If false, use <code>SMUSER</code>.

Configuring IBM WebSEAL single sign-on (SSO) authentication

EMC Documentum can integrate with IBM WebSEAL, its SSO solution, or any other SSO solution supported by IBM WebSEAL.

IBM WebSEAL documentation contains more information on installing and configuring IBM WebSEAL and on enabling IBM WebSEAL SSO authentication.

Prerequisites

- Set the precedence of authentication schemes in the `com.documentum.web.formext.session.AuthenticationSchemes.properties` file. *EMC Documentum Web Development Kit Development Guide* contains more information.
- Install the IBM WebSEAL server on a machine, and create an HTTP or HTTPS junction that links the IBM WebSEAL server to Documentum Administrator.

The *IBM WebSEAL documentation* contains more information on installing and configuring the IBM WebSEAL web server.

- Deploy Documentum Administrator on the application server machine, and connect to a Content Server that has been configured for IBM WebSEAL SSO authentication. *EMC Documentum Platform and Platform Extensions Installation Guide* and *EMC Documentum Content Server Administration and Configuration Guide* contains more information on configuring Content Server for IBM WebSEAL SSO authentication.

Configurations in custom/app.xml file to enable IBM WebSEAL authentication

Set the value of the `user_header` tag to **iv-user**, within the authentication tag:

```
<authentication>
  <webseal_config>
    <user_header>iv-user</user_header>
  </webseal_config>
</authentication>
```

Note: Copy the `user_header` element into the authentication tag of the `custom/app.xml` file.

Configuring Kerberos authentication

Kerberos SSO authentication scheme is used to authenticate the user who wants to log in to Documentum Administrator.

EMC Documentum supports Kerberos secure Single-Sign-On (SSO) using Microsoft Active Server Domain Services for Kerberos Key Distribution Center (KDC) services in the following ways:

- In a single domain.
- In one-way and two-way trusts between multiple domains in the same forest only; that is, cross-forest trusts are not supported.

Kerberos-based single sign-on authentication in Documentum Administrator

When Kerberos-based Single Sign-On Authentication is enabled on Documentum Administrator, users of Documentum Administrator are automatically authenticated and logged in to the repository using their credentials stored in the user's private credential area on the Windows platform.

Prerequisites

- Deploy Documentum Administrator on the application server machine, and connect to a Content Server that has been configured for Kerberos SSO authentication. *EMC Documentum Platform and Platform Extensions Installation Guide* and *EMC Documentum Content Server Administration and Configuration Guide* contains more information on configuring Content Server for Kerberos SSO authentication.
- Install a supported browser on the client machine.
- Register Documentum Administrator as a Service Principal in the Key Distribution Center (KDC). The steps are discussed in more detail in the following sections but are considered prerequisites because they may require customer's infrastructure team involvement.
- On a Windows Server host, ensure that the following key and value have been added to the registry for Java to use to acquire additional service tickets:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
Value Name: allowtgtsessionkey
Value Type: REG_DWORD
Value: 0x01
```

Configurations in custom/app.xml file to enable Kerberos authentication

Perform the configurations specified in this section, in the <enabled>, and <domain> tags within the <authentication> tag, and copy the configurations into the custom/app.xml file.

Enabling Kerberos SSO authentication in Documentum Administrator

An application level setting is provided in custom/app.xml within the <authentication> tag to enable or disable Kerberos-based SSO authentication. The default value defined for the <enabled> tag in the <kerberos_sso> element is "false". Set the <enabled> tag to *true* to enable Kerberos SSO authentication.

```
<kerberos_sso>
  <enabled>true</enabled>
</kerberos_sso>
```


Configuring the Kerberos domain name

An application level tag is provided to specify the Kerberos domain, within the <authentication> tag. Enter the domain name in the <domain> tag.

```
<kerberos_sso>
  <domain><domain_name></domain>
</kerberos_sso>
```

Configuring Kerberos fallback

The Kerberos SSO Authentication Scheme provides the option to fall back to the default login mechanism to the web-application, on failure conditions. Set the <docbase_login_fallback> tag in the <kerberos_sso> tag in custom/app.xml, to support the default login to the web-application, as follows:

```
<docbase_login_fallback>true</docbase_login_fallback>
```

The default value of the <docbase_login_fallback> tag is *false*.

Copy the <docbase_login_fallback> element into the <kerberos_sso> tag in custom/app.xml.

Sample Kerberos configuration in app.xml

The following code snippet is an example of the final configuration for Kerberos in app.xml.

Example 5-1. Code snippet in custom/app.xml file to enable Kerberos authentication

```
<authentication>
  <!-- Kerberos SSO authentication scheme configuration -->
  <kerberos_sso>
    <enabled>true</enabled>
    <browsers>
      <windows>
        <ieversions>8.0,9.0,10.0,11.0</ieversions>
        <firefoxversions>10.0</firefoxversions>
      </windows>
    </browsers>
    <!-- Enable login fall back to DocbaseLogin scheme -->
    <docbase_login_fallback>false</docbase_login_fallback>
    <!-- Mandatory configuration: Provide the kerberos realm / domain name. -->
    <domain>WDKBLR.COM</domain>
  </kerberos_sso>
</authentication>
```

Copy the <authentication> tag from wdk/app.xml file into the custom/app.xml file.

Preparing Documentum Administrator and the browser to meet Kerberos SSO setup requirements

This section discusses the setup requirements to enable Kerberos single sign-on authentication in Documentum Administrator. Ensure that the client machine is already configured to use Kerberos authentication before you prepare the system for enabling Kerberos-based authentication.

Create user account for Documentum Administrator in the active directory

You must register Documentum Administrator as a Kerberos principal in the active directory to enable the Documentum Administrator application to participate in Kerberos authentication. A Kerberos principal is a regular account on an Active Directory. The name of the principal can be something like this "name@YOUR.REALM". The realm name follows the "@" character in the principal. The principal represents the Documentum Administrator application service in the Kerberos realm.

Define a Service Principal Name for Documentum Administrator and create KeyTab file

After defining the SPN for the application server (on which Documentum Administrator is deployed), the administrator must create a keytab (key table) file for Documentum Administrator. Documentum Administrator requires the keytab file to authenticate itself to the Key Distribution Center (KDC).

The administrator must use the ktpass command-line tool to register the SPN as a security principal in the Windows Server Active Directory and to create a KeyTab file on the KDC. This `ktpass.exe` is bundled with Windows 2008 Resource Toolkit package and must be installed separately. Run `ktpass.exe` on the Active Directory Server machine and when the keytab file is generated move it to the `da_installation/WEB-INF` folder on the application server machine.

```
ktpass /pass <password> -out <user-name>.keytab -princ <SPN> -crypto  
AES128-SHA1 +DumpSalt -ptype KRB5_NT_PRINCIPAL  
/mapOp set /mapUser <user-name>
```

Example 5-2. You can run the ktpass command with the following parameters:

```
ktpass /pass <password> -out da.keytab -princ  
HTTP/da.dctmlabs.com@DCTMLABS.COM -crypto AES128-SHA1 +DumpSalt  
-ptype KRB5_NT_PRINCIPAL /mapOp set /mapUser da
```

This command generates the `da.keytab` file on the Active Directory machine. Copy this file to the `da_installation/WEB-INF` folder on the application server machine.

Configuring the client browser to use the SPNEGO protocol

Configure your browser to use the SPNEGO protocol. In Internet Explorer, ensure that the **Enable Integrated Windows Authentication (requires restart)** option is selected and click **OK**. In Firefox, double-click the **network.negotiate-auth.trusted-uris** and **network.negotiate-auth.delegation-uris** preferences.

In Windows, the Data Encryption Standard (DES) encryption type (security settings) for Kerberos is disabled by default. If you log in to Documentum Administrator from a client computer having Windows as the operating system, you should enable the following:

- DES_CBC_CRC
- DES_CBC_MD5
- RC4_HMAC_MD5
- AES128_HMAC_SHA1
- AES256_HMAC_SHA1

The *Microsoft Windows documentation* contains the instructions.

Creating JAAS configuration file

Apache Tomcat, Oracle WebLogic, and VMware vFabric tc Server use the JAAS configuration file to obtain the Login context. The `KerberosSSOAuthenticationScheme` class uses the Java JAAS and GSS-API to perform Kerberos authentication. The administrator must create the JAAS configuration file in the `da_app_root_directory/WEB-INF` folder; for example, `da_app_root_directory/WEB-INF/krb5Login.conf`.

Create the JAAS configuration file as follows:

```
<loginContext>
{
  <LoginModule> required
  principal="<SPN>"
  realm="<REALM>"
  refreshKrb5Config=true
  noTGT=true
  useKeyTab=true
  storeKey=true
  doNotPrompt=true
  useTicketCache=false
  keyTab="<DAuser_keytab_path>";
};
```

where:

<loginContext>	<p>Corresponds to the DA SPN. You replace separator characters with hyphen characters and omit the @REALM segment in the SPN. For example, the following LoginContext is derived from the corresponding SPN:</p> <ul style="list-style-type: none"> • LoginContext: <code>HTTP-wdkapps-wdkblr-com</code> • SPN: <code>http/wdkapps.wdkblr.com@WDKBLR.COM</code> <p>Note: Make sure that the SPN in the JAAS configuration matches the SPN defined in <code>web.xml</code>.</p>
<LoginModule>	<p>Specify the Kerberos login module to be used to perform user authentication:</p> <ul style="list-style-type: none"> • For single-domain support only: <code>com.sun.security.auth.module .Krb5LoginModule</code> • For both multi- and single-domain support: <code>com.dstc.security.kerberos.jaas .KerberosLoginModule</code> <p>Note: This module is the Quest KerberosLoginModule.</p>
<SPN>	<p>The DA SPN.</p> <p>For example, for single-domain support: <code>http/wdkapps.wdkblr.com@WDKBLR.COM</code></p> <p>For multi-domain support, instead of appending the domain name to the SPN, use the <code>realm</code> property to specify the domain name.</p>
<REALM>	<p>(Multi-domain support only) The realm name. For example: <code>WDKBLR.COM</code></p>
<DAuser_keytab_path>	<p>The path to the DA user account's *.keytab file in the WEB-INF folder of Apache Tomcat. For example: <code><da_app_root>/WEB-INF/xxx.keytab</code></p>

Creating a configuration file for the application server to connect to the KDC server

To specify the KDC server to which the application server connects, create a configuration file in the %WINDIR% directory of the Windows operating system or the /etc folder of the UNIX and Linux operating systems. The names of the configuration files are `krb5.ini` (Windows) and `krb5.conf` (UNIX and Linux) respectively. Refer to the following examples.

Example 5-3. Create the configuration file with the following contents to specify Data Encryption Standard (DES) as a permitted encryption type:

```
[libdefaults]
default_realm = WDKBLR.COM
forwardable = true
ticket_lifetime = 24h
clockskew = 72000

default_tkt_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1
default_tgs_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1
permitted_enctypes = des-cbc-md5 des-cbc-crc des3-cbc-sha1

[realms]
    WDKBLR.COM= {
        kdc = WDKWIN5175.WDKBLR.COM
        admin_server = WDKWIN5175.WDKBLR.COM
    }
```

The following example is to specify the Advanced Encryption Standard (AES) as a permitted encryption type along with the DES.

Example 5-4. Create the configuration file with the following contents to specify both DES and AES as permitted encryption types:

```
[libdefaults]
default_realm = <Kerberos_domain_name>
forwardable = true
ticket_lifetime = 24h
clockskew = 72000

default_tkt_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
default_tgs_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
permitted_enctypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc

[realms]
    <Kerberos_domain_name>= {
        kdc = <KDC_server_address>
        admin_server = <KDC_server_address>
    }
```

Modify the Windows configuration file with the following details:

- Specify the Kerberos domain name as the `default_realm`.
- The `realms` section points to the KDC server.

Application Server-specific configurations

While configuring the application servers for Kerberos authentication the following application server-specific configurations are a prerequisite. Perform the following configurations that are specific to your application server, on which Documentum Administrator is deployed.

Apache Tomcat

In `Tomcat_home_directory/bin/Catalina.bat` or `catalina.sh`, set the following JAVA options:

```
set JAVA_OPTS=% JAVA_OPTS % -Djava.security.krb5.conf=<location of krb5.ini>
-Djava.security.auth.login.config=<location of krb5Login.conf>
-Djavax.security.auth.useSubjectCredsOnly=false
```

Oracle WebLogic

In `WebLogic_home_directory\user_projects\domains\your_domain\bin\setDomainEnv.cmd` file or the `setDomainEnv.sh`, set the following JAVA options:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Xms256m -Xmx1024m -Xdebug -Xnoagent
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
-Djava.security.krb5.conf=<location of krb5.ini>
-Djava.security.auth.login.config=<location of krb5Login.conf>
-Djavax.security.auth.useSubjectCredsOnly=false
```

Note: The default location of the `krb5.ini` file is `%WINDIR%` (Windows).

IBM WebSphere

- In `WebSphere_home_directory\AppServer\profiles\AppSrv01\properties\wsjaas.conf`, add the following configuration:

```
HTTP-hostName-realm_Name { com.ibm.security.auth.module.Krb5LoginModule
    required debug=true credsType="both" useKeytab="file:fullPathToKeytabfile"
    principal="HTTP/hostname.realmName"; };
```

- Create a configuration file to specify the KDC server to which the application server should connect, in the `%WINDIR%` (Windows) or in `/etc/krb5` (AIX). The names of the configuration files are `krb5.ini` (Windows) and `krb5.conf` (AIX). To support Advanced Encryption Standard (AES) in the Websphere Application Server, specify `aes128-cts-hmac-sha1-96` as a permitted encryption type.

Example 5-5. Both DES and AES as permitted encryption types

```
[libdefaults]
    default_realm = WDKBLR.COM
    forwardable = true
    ticket_lifetime = 24h
    clocks skew = 72000

    default_tkt_enctypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-sha1
    des-cbc-md5 des-cbc-crc
    default_tgs_enctypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-sha1
```

```

des-cbc-md5 des-cbc-crc
permitted_encetypes = aes128-cts aes128-cts-hmac-sha1-96 des3-cbc-sha1
des-cbc-md5 des-cbc-crc

[realms]
    WDKBLR.COM= {
        kdc = WDKWIN5175.WDKBLR.COM
        admin_server = WDKWIN5175.WDKBLR.COM
    }

```

Cross-frame scripting configuration

To protect Documentum Administrator from cross-frame scripting issues, enable `<frame_bursting>`.

```

<frame_bursting>
  <enabled>true</enabled>
</frame_bursting>

```

Starting Documentum Administrator

Before you test the deployment, ensure that Documentum Administrator is started in the application server. The documentation on each web application server contains information on starting the application.

To verify Documentum Administrator deployment and configuration:

1. Open a browser window and type the following URL:

```
http://host_name:port_number/virtual_directory
```

where:

- *host_name* is the host where the application server is installed. If the browser is on the application server machine, substitute `localhost` for *host_name*. For example: **http://localhost**.
- *port_number* is the port where the application server listens for connections.
- *virtual_directory* is the virtual directory for your application.

For example, if the application server host is named `iris`, the port number is 8080, and the application virtual directory is `da`, the URL is **http://iris:8080/da**.

2. Use Documentum Administrator application to log in to a repository.

If the login succeeds, the application is correctly deployed and configured.

Maintenance and procedures

After the installation, it is essential to follow a maintenance/procedure checklist for maximum system performance and stability.

Many of the maintenance procedures and jobs are configured or accessed through Documentum Administrator:

- Server and Repository configurations
- LDAP configuration
- Users, Groups, Roles
- Security (ACLs)
- Storage (Locations, Storage, and Filestores)
- Index Agent's failed index list should be understood and resubmitted, if necessary

Logs to monitor

It is highly recommended to check all logs periodically for errors and warnings.

Application Server

- Name: `stdout_YYYYMMDD.log` (for example, `stdout_20090218.log`)
- Location: Application Server logs directory
- Purpose: Warnings and errors from Documentum Administrator and TBOs

Content Server repository

- Name: `DocbaseName.log`
- Location: `$DOCUMENTUM\dba\log`
- Purpose: Repository startup output and any warnings or errors

Java Method Server

- Name: `access.log` and `DctmServer_MethodServer_DocbaseName.log`
- Location: `%JBOSS_HOME%\server\DctmServer_MethodServer\logs`
- Purpose: Access and status of the Java Method Server

Index Server

- Name: `access.log` and `DctmServer_IndexAgent.log`
- Location: `%JBOSS_HOME%\domains\DctmDomain\servers\DctmServer_IndexAgent\logs`
- Purpose: Access and status of index agent

Disk space management

The Content Server has a state of the repository job (`dm_StateOfDocbase`) which monitors this. Also, the data drive should be monitored.

Monitor the following:

- SQL Server transaction log
- Webtop cache files
- Index data drive
- Database maintenance and logs
- Disk space
- Transaction logs
- CPU and RAM usage patterns

Jobs

Some of the jobs discussed in this section are not active OOTB. They have to set to active and started on a schedule. Ensure to set the run times so that they do not conflict other jobs and backup schedules.

- `dm_ContentWarning`: Provides warnings for low availability on DM content/fulltext disk devices.
- `dm_LogPurge`: Removes outdated server/session, and job/method logs method.
- `dm_StateOfDocbase`: Lists the repository configuration and status information. Also, displays the number of documents and total size of content.
- `dm_AuditMgt`: Removes old audit trail entries A key parameter is the cutoff in days, basically how many days worth of audits to keep.
- `dm_QueueMgt`: Deletes dequeued items from `dm_queue`.
- `dm_UpdateStats`: Updates RDBMS statistics and reorganizes tables (if RDBMS supports).
- `dm_ConsistencyChecker`: Checks the consistency and integrity of objects in the repository.
- `dm_DataDictionaryPublisher`: Publishes the data dictionary information.
- `dm_LDAPsynchronization`: Used for one-way synchronization of LDAP users and groups to Docbase Method.
- `dm_FTStateOfIndex`: State of Index `dm_FTIndexAgentBoot` Boot Index Agents Method.
- `dm_GwmTask_Alert`: Sends email alert if task duration exceeds.
- `dm_GwmClean`: Cleans all the orphan decision objects.

DQL queries

This section discusses the DQL queries to be run to check on audit trails and `dmi_queue_items`.

The following statements are some of the DQLs to determine the number of audit trails and queue items that were in the repository:

```
Select count(*) from dmi_queue_item  
Select count(*) from dm_audittrail
```

Network connectivity interruption

If any network interruption occurs, then service logs should be checked for compromised activity. The Content Server and Tomcat server may need to be restarted. The logs of the application and Content Servers should be periodically monitored for errors and warnings.

RAM and CPU Utilization maxed out

If RAM is filled or CPU utilization is maxed out then the service responsible should be checked. If the service is a Documentum service, it should be restarted and root cause should be determined. Utilization should be monitored and any anticipated spikes in use or additional services need to be load tested and analyzed. If the application server's performance is slow and the concurrent users reach EMC's limit of 20, EMC recommends adding a second application server.

Sessions to monitor

This section discusses the different ways to monitor sessions.

- Documentum Administrator: **Administration > User Management > Session**
- DQL:
 - execute show_sessions: To display all active and inactive sessions
 - execute list_sessions: To display active sessions
- DocBasic ebs script: Set this script at a command line prompt to output how many active and inactive sessions are current on the Content Server. Set the interval between output and how many loops to run.

Security and Server access maintenance

You can perform the following for the security and server access maintenance:

- Test users and test content should be deleted out of production.
- The database schema owner account should be locked down.
- The Documentum install owner `dmadmin` should be locked down.
- Only scheduled, authorized access to the production should be allowed for all servers of the system.
- Repository audit trails should be configured for certain events, such as deleting of content.

Improving Performance

This section lists the guidelines that can significantly improve performance of your web application. The system sizing spreadsheet is available on EMC Online Support.

- Improving search query performance
Set <showfolderpath> to false in the search component to speed queries.
- Disable tracing
Turn off tracing to improve performance. Navigate to the page wdk/tracing.jsp and deselect all tracing flags.
- Set `dfc.diagnostics.resources.enable` to *false* in the `dfc.properties` file unless you are using the DFC diagnostics. This setting uses a significant amount of memory.

Java EE Memory Allocation

If the memory allocated to the Java EE server Java virtual machine (VM) is not correctly set, the VM will spend a lot of time destroying Java objects, garbage collecting, and creating new objects. To change the memory allocation, use a setting similar to the following in the Java arguments in the Java EE server start script that you use to start your application server:

```
-Xms512m -Xmx512m -verbose:gc
```

Element	Description
-Xms512m	Starting memory heap size, in megabytes. In general, increased heap size increases performance up until the point at which the heap begins swapping to disk.
-Xmx512m	Maximum Heap size. For a single VM, Sun recommends that you set maximum heap size to 25% of total physical memory on the server host to avoid disk swapping. Increased heap size will increase the intervals between garbage collection (GC), which thus increases the pause time for GC.
-verbose:gc	Turns on output of garbage collection trace to standard output. Increased Java memory settings will increase the amount of time before a major garbage collection takes and will also increase the amount of time that garbage collection takes to execute. Garbage collection is the greatest bottleneck in the application, and all application work pauses during garbage collection.

Garbage collection tracing has the following syntax:

```
[GC 776527K->544591K(1040384K), 0.4283872 secs]
```

The trace can be interpreted as follows:

Element	Description
GC	GC indicates minor garbage collection event, Full GC indicates full garbage collection

Element	Description
776527K	Amount of total allocated memory at start of minor collection
544591K	Amount of total allocated memory at end of minor collection
1040384K	Amount of total memory on host
0.4283872 secs	Time in seconds to run garbage collection

Monitor memory usage by the Java process in the Windows task manager to determine whether your memory allocations are optimum. Allocated memory as shown in consecutive GC traces continues to grow until full garbage collection occurs. Full garbage collection takes much longer than minor garbage collection, often on the order of 10 times as long.

The following table describes some memory troubleshooting inferences that can be drawn from garbage collection.

Symptom	Reason
Frequent full GC, starting point higher after each full GC, decreasing number of GC between full GC	Total memory too small, or memory leak
Garbage collections take too long (GC 1 sec, full GC 5 sec), server cannot create new threads	Too much memory allocated to JVM

Java EE servers also have configurable settings for thread management which can significantly affect performance. The symptom of insufficient threads is that, as the number of users increases, performance degrades without increased CPU usage. Some users will get socket errors. In Tomcat, the log catalina.log shows that all threads up to maxProcessors have been started, and new requests are rejected with "Connection Reset By Peer". In WebLogic, the execute queue shows waiting threads (0 idle threads, with queue length growing).

The symptom of too many threads is excessive context switching between live threads and degraded response time.

Your application server documentation contains more information on these settings.

Preferences

User preferences are stored as cookies and written to the repository. Since cookies are passed back and forth with every request and response, there is a small increase in network traffic.

The configuration lookup methods `lookupString`, `lookupInteger`, and `lookupBoolean` have an optional parameter `consultPreference`. Set to false to look up a configuration value from the component definition and bypass a lookup of the user preference when the lookup is not needed.

Browser History

The number of history pages maintained on the server for each window or frame is set by the `requestHistorySize` flag in the file `FormProcessorProp.properties`, which is located in

WEB-INF/classes/com/documentum/web/form. The default value is 3. If the value is empty or zero, then history is maintained indefinitely. This setting could significantly affect performance. Decrease the memory footprint per user by setting this value lower. If you set it higher, it will consume more memory.

Too many form history objects can use up memory. Set the upper limit for the number of objects as the value of `maxNoOfFormHistoriesThreshold` in `FormProcessorProp.properties`. The default value is 50. A message will be displayed if the user tries to navigate past the maximum number of pages in history.

Memory that is allocated to maintaining browser history is managed more efficiently on the Java EE server if you generate framesets and frames using the `<dmf:frameset>` and `<dmf:frame>` tags. *EMC Documentum Web Development Kit Development Guide* contains more information.

Value Assistance

Performance is affected by the number of value assistance queries to be displayed in the properties component and in other components that display a set of properties. Do the following to enhance this performance:

- For each value assistance query, use Composer or Documentum Application Builder to turn on the option to allow caching.
- Turn on client persistent caching in `dfc.properties`, which is located in WEB-INF/classes:

```
dfc.cache.enable_persistence = T
```
- Index the associated attributes in Content Server.

Search Query Performance

Set `<displayresultspath>` to false in your custom search component definition to speed all queries. This suppresses the query for folder path of each object.

In advanced search, you can add a checkbox for case-sensitive search for non-indexed repositories. Set the `casevisible` attribute on the search controls to true. Set the default match case as the value of the element `<defaultmatchcase>` (true | false) in `wdk/config/advsearchex.xml`. Case-sensitive queries perform faster.

High Latency and Low Bandwidth Connections

Two filters are available to improve performance in high latency or low bandwidth networks. The filters are defined as servlet filters in WEB-INF/web.xml. They are turned on by default. The filters are as follows:

- Response compression filter (`CompressionFilter`)
 Compresses text responses by mapping requests for `*.jsp`, `*.css`, `*.js`, `*.htm`, `*.html`, and the component dispatcher servlet. If the request accept- header indicates that the browser accepts compression, the filter swaps the output stream for a compressed stream in either gzip or deflate

compression formats, depending on which format is accepted by the browser as indicated by the Accept- request header.

The configurable value for this filter, init-param compressThreshold, is a size in KB or MB that sets the threshold file size at which output will be compressed. Compression does not decrease the size of the stream for small inputs. Additional, high-bandwidth networks may show improvement for only very large files (hundreds of KB). A value of 3kb indicates that files 3 KB or larger will be compressed.

Additionally there are init-params for turning on compression filter debugging and excluding specific JSP pages from compression filtering.

Limitation: There is an unknown CPU cost for the compression.

- Cache control (ClientCacheControl)

Limits the number of requests by telling the client browser and any intermediary caches such as caching proxies to cache static elements such as *.gif, *.js, and *.css files, by adding a Cache-Control response header. After the browser has received a response with this header, it will not re-get the content until the maximum age or until the content is cleared manually from the browser cache.

The configurable value for this filter, init-param Cache-Control, is the maximum age in seconds of the static content before revalidation, for example, max-age=86400 (one day).

Add URL patterns to specify the file types that will be cached. In the following example, *.gif files are cached for up to two days:

```
<filter>
  <filter-name>ClientCacheControl</filter-name>
  <filter-class>com...ResponseHeaderControlFilter</filter-class>
  <init-param>
    <param-name>Cache-Control</param-name>
    <param-value>max-age=172800</param-value>
  </init-param>
</filter>
</filter>
<filter-mapping>
  <filter-name>ClientCacheControl</filter-name>
  <url-pattern>*.gif</url-pattern>
</filter-mapping>
```

Note: Safari browser does not apply this header. IE does not support both the cache-control and compression mechanisms at the same time.

Tracing for these filters can be enabled through the standard tracing mechanism (TraceProp.properties) or by adding the debug <init-param> element to the application deployment descriptor (WEB-INF/web.xml).

Example 5-6. Enabling tracing of filters in WEB-INF/web.xml

```
<filter>
  <filter-name>CompressionFilter</filter-name>
  <filter-class>com.documentum.web.servlet.CompressionFilter</filter-class>
  <init-param>
    <param-name>compressThreshold</param-name>
    <param-value>3kb</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>
```

Qualifiers and Performance

Each qualifier that is defined in the application slows performance the first time a component is called. Navigation components must evaluate qualifiers for each action in the component JSP page. To improve performance, remove from your custom app.xml file the qualifiers that your application does not need. (The application qualifier is required.) In the following example from an app.xml file in the custom directory, only the type qualifier is used by a custom application. The app qualifier is required for all applications. No components or actions can be scoped to role in this example, because the role qualifier is not defined for the application.

```
<qualifiers>
  <qualifier>com.documentum.web.formext.config.DocbaseTypeQualifier
</qualifier>
  <qualifier>com.documentum.web.formext.config.AppQualifier
</qualifier>
</qualifiers>
```

For better performance, your qualifier should implement the `IInquisitiveQualifier` interface. At startup, this interface is used to inform the qualifier of all relevant scopes defined in the action and component definitions. The qualifier can return an empty scope value that is cached, when the runtime context is not relevant.

Import Performance

You can limit the number of files that can be imported by a user during a single import operation. This configuration setting is the `<max-import-file-count>` element with a default of 1000 in the `importcontainer` component. Extend this component definition to configure a different maximum value.

Certain environments have forward or reverse proxy web servers that do not support HTTP 1.1 chunking, which is used by UCF for content transfer. For those environments, you must configure UCF to use alternative chunking, and you can tune the chunk size for the web server. In general, the default chunk size works best for large file transfers. Smaller chunk sizes may enhance performance for small (less than 1MB) files but degrade performance for large files. *EMC Documentum Web Development Kit Development Guide* contains more information.

Load Balancing

WDK applications can be load balanced using network load balancers. Session "stickiness" (or affinity) must be used. That is, once a session has been established between a browser and a back-end application server then all subsequent traffic from that browser must be routed to that server by the load balancer for the duration of the session. When using Load Balancer in conjunction with Network Location for BOCS product, ensure that the load balancer is configured to pass the client IP in the HTTP request, so that the web application picks up the client IP and map to the correct network location.

Because content transfer is disk-intensive, best performance spreads the I/O of the WDK content directory over a striped disk volume.

Modal Windows and Performance

Modal windows provide a performance enhancement in web applications that use several frames. With a modal window, other frames do not need to refresh after the modal frame closes. *EMC Documentum Web Development Kit Development Guide* contains more information.

Troubleshooting deployment

Wrong JRE used for application server

If the application server host has multiple JREs on the system, the application server can use the wrong JRE. Check your application server documentation for instructions to use the correct JRE with your application server. For example, the Apache Tomcat application server uses a `JAVA_HOME` environment variable. This variable value is specified in the application startup batch file `catalina.bat` or in the `service.bat` file for Windows services.

If the application server uses the wrong JRE, Apache Tomcat displays the following error:

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

No global registry or connection broker

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

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

To fix this error, provide the correct BOF registry connection information in the `dfc.properties` file or do not provide any connection information at all. *EMC Documentum Platform and Platform Extensions Installation Guide* contains information on enabling a repository as a global registry.

No connection to repository

If a connection broker is not specified in the `dfc.properties` file of the Documentum Administrator WAR file, the application server log contains the following error during application initialization:

```
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:432)
Caused by: DfDocbrokerException:: THREAD: main; MSG: [DFC_DOCBROKER_REQUEST_FAIL
ED] Request to Docbroker "10.8.3.21:1489" failed; ERRORCODE: ff; NEXT: null
```

To establish a connection to repositories, Documentum Administrator must have information about the available connection broker. Enable the connection in the `dfc.properties` file.

If the repository that is specified as the global repository is down, the following message appears:

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

Login page incorrectly displayed

If the login page displays several login buttons, the browser does not have the Java plug-in installed. Download and install the Java plug-in for the browser.

If the login page displays several controls with the same label, you have not turned off tag pooling in the application server.

Slow performance

[Improving Performance, page 379](#) contains the information.

Out of memory errors in console or log

Verify that you have allocated sufficient RAM for the application server VM. Set the Java memory allocation.

The `java.lang.OutOfMemoryError: PermGen space` error is common when `MaxPermSize` is set too low.

Slow display first time

The application server must compile a JSP the first time it is accessed. It is much faster on subsequent accesses. If you have tracing turned on, or if you have a large log file (of several megabytes), the browser response time decreases dramatically.

DFC using the wrong directories on the application server

If you have not specified content transfer directories in the `dfc.properties` file, DFC looks first for global environment variables that set directory locations.

Tag pooling problem

If you have not properly disabled tag pooling in the application server, you see several instances of the same control on the login page.



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

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

UCF client problems

If the error message `Compatible Java Run time environment is not installed` is displayed on a non-Windows client, verify that you have installed a certified version of the JRE on the client. UCF uses this version, which does not interfere with the browser VM. It is used for non-UCF applets.

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

- For UCF timeouts, check whether anti-virus software on the application server is monitoring port 8080 or the application server port that is in use. Turn off monitoring of the application server port.
- For slow UCF downloads, ensure that virus scanning within zip files is not turned on.
- Ensure that the user has a supported JRE version on the machine to initiate UCF installation. To verify the presence and version of a JRE, you can point the client browser to a Java tester utility.
- Verify if the process from the launch command is running: Open the browser Java console look for invoked runtime: `... connected, uid: ...` A UID indicates successful connection to the UCF server.
- Check the application server console for errors on the UCF server.
- Restart the browser and retry the content transfer operation.
- Kill the UCF launch process and retry the content transfer operation.
- If UCF operations still do not launch, delete the client UCF folder located in `USER_HOME/username/Documentum/ucf`.
- Search the client system for files that start with `ucfinit.jar`- and delete them.

Connection issues between a Federated Search server and IPv6 clients

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

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

Max Sessions error

Before restarting the application server, use the Documentum Administrator to find the current "active" and "inactive" users sessions in the repository. Try reducing the session timeout value in the application server to see if the inactive sessions get cleared out faster.

Documentum Foundation Classes

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

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

Before you install DFC

This section describes the steps you must take before installing DFC. Any EMC Documentum client product that uses DFC installs DFC. If both DFC and its client applications are compiled with the same JAVA version, you can upgrade DFC without upgrading the client applications.



Caution: If you previously installed additional classes on top of DFC under the DFC program root directory, check whether those classes are still in place after you upgrade DFC. For example, you may have customized your system and created custom classes and installed them in the DFC shared directory after installing DFC.

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

Note: Before running the DFC standalone installer, please install the following libraries and their dependencies:

- libXp (i686)
- libXi (i686)
- libXtst (i686)
- libXt (i686)

DFC relies upon certain environment variables. Ensure that these variables have correct values. [Establishing the environment for DFC, page 391](#) explains how to set the environment variables that DFC relies upon.

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

Where to install DFC

You can install DFC on:

- A middle-tier system.

For example, to support WDK or Content Server methods.

- An end user's computer.

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

Note: Using DFC with an application server may further restrict the supported versions.

Java 2 Security

If you plan to use the Java 2 security, archive the policy file that you create at the startup of your Java Virtual Machine (JVM). For a standalone installation, you will specify the file in the command line startup arguments for your JVM. If you are deploying on an application server, refer to the application server documentation for the specifics of specifying Java 2 security policies. The DFC example policy template file is named `dfc.example.java.policy`, and is found in the `dfc.jar` file installed in the `DOCUMENTUM_SHARED` directory (by default, in Windows systems, `C:\Program Files\Documentum\Shared`).

If you deploy DFC on WebSphere Application Server, you must set the `dfc.security.enforce_existing_policy` JVM runtime system property before policy configuration.

Name	Description	Type	Required or Not	Default
<code>dfc.security.enforce_existing_policy</code>	<p>Specifies whether to enforce the existing security policy settings or not.</p> <p><code>true</code>: DFC enforces the existing security policy settings. And thus, policy configuration is skipped.</p> <p><code>false</code>: DFC does not enforces the existing security policy settings. And thus, you can initialize policy configuration to update the security settings.</p>	boolean	Yes	false

Removing old files

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

Removing old DFC

You can install this release of DFC directly over DFC version 5.1 or later.

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

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

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

[Uninstalling DFC, page 395](#) explains how to remove DFC.

Whether to upgrade client programs

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

In order to run more than one version of DFC on a Linux system, you must arrange to run the different DFC versions in different processes. Set the environment variables described in [Establishing the environment for DFC, page 391](#) and install the different versions of DFC in locations that you can distinguish from one another by those environment settings.

Establishing the environment for DFC

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

[Table 16, page 393](#) lists the environment variables that DFC relies upon. To set the variables:

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

The DFC installation program for Windows sets environment variables. The only additional setting you need to make is to add jars to the classpath if you need to refer to DFC classes and interfaces in your Java programs. [Locations of DFC classes, page 393](#) provides more details about what to place on the classpath.

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

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

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

The following topics describe the considerations in setting up the DFC environment for Windows or Linux.

Defining file system locations for DFC components

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

DFC program root directory

DFC installs program files under the program root directory.

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

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

DFC user root directory

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

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

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

Directory for shared libraries

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

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

On Linux systems the installation program uses the `dfc` subdirectory of the program root directory. You must place the full path of this directory onto the library path. The library path environment variable is `LD_LIBRARY_PATH` for Linux.

Environment variables can be set on Linux systems using the *setenv* script. The script can be found at `$DOCUMENTUM_SHARED/dfc/set_dctm_env.sh` (.csh). You can source this file to properly set the environment variables from [Table 16, page 393](#).

Directory for DFC configuration files

The installation program creates the config directory to store configuration files. [Using the DFC config directory, page 394](#) provides information about DFC configuration files. The installation program creates the config directory under the program root directory on Linux systems and under

the user root directory on Windows systems. For DFC to operate properly, the full path to the config directory must appear in the classpath.

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

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

Locations of DFC classes

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

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

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

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

Setting environment variables

DFC uses several environment variables to find its components. [Defining file system locations for DFC components, page 392](#) describes the file system locations that the environment variables point to. On Windows systems, the installation program asks you for the information that it uses to set these variables. On Linux systems, you must set these variables before you run the installation program. [Table 16, page 393](#) lists these environment variables and summarizes the ways that DFC uses them.

Table 16. Environment variables that DFC uses

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

Variable	How DFC uses it	Windows value (installation program sets)	Unix value (you set)
LD_LIBRARY_PATH	Find the directory containing DFC shared libraries on Linux and SOLARIS	Not used by Windows systems	Add \$DOCUMENTUM_SHARED/dfc
LIBPATH	Find the directory containing DFC shared libraries on AIX	Not used by Windows systems	Add \$DOCUMENTUM_SHARED/dfc
DFC_DATA	Documentum has deprecated this variable.	Directory for DFC configuration files, page 392 provides information about what you should do instead of using this variable.	
DOCUMENTUM	Determine the full path to the user root directory	Not used by Windows systems	Specify a value before installing DFC
CLASSPATH	Allow Java runtime to find dctm.jar and, the DFC config directory. See Locations of DFC classes, page 393 for information about making DFC classes available to the javac compiler	Attach (with appropriate separator characters) the full paths of dctm.jar and the config directory (for example, C:\Documentum\Shared\dctm.jar and C:\Documentum\config)	Add \$DOCUMENTUM_SHARED/dctm.jar and \$DOCUMENTUM_SHARED/config

Using the DFC config directory

The DFC config directory contains Java properties files that control the behavior of DFC. The installation program creates the config directory if it does not already exist. [Table 17, page 394](#) describes the files in the config directory.

Table 17. Configuration files for DFC

File	Description
dfc.properties	Current configuration options for DFC.
dfcfull.properties	Template containing all possible configuration options. Do not modify this file. Copy sections into dfc.properties as necessary.

File	Description
log4j.properties	Current configuration options for the log4j instance that underlies the unified logging system. <i>EMC Documentum Foundation Classes Development Guide</i> explains the logging system.
dbor.properties	Registry for pre-5.3 business objects. Do not edit this file. <i>EMC Documentum Foundation Classes Development Guide</i> contains information about how to use this file.

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

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

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

A backslash precedes each colon, backslash, or space.

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

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

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

After installation, the `dfc.properties` file contains the key `dfc.data.dir`. The corresponding value is the full path to the DFC data directory.

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

Uninstalling DFC

This section explains how to remove DFC. [Removing old DFC, page 391](#) lists the situations in which you must remove an old version of DFC before installing the current version. Installation program uses InstallAnywhere.

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

Uninstalling from Windows

1. Change the startup setting to manual for each service that uses DFC.
This prevents such services (for example, an application server) from restarting themselves after you reboot and locking DFC again before you can uninstall it.
2. Use the control panel’s Add/Remove Programs facility to remove Documentum DFC Runtime Environment.

3. If prompted to do so, reboot the system.
4. Restore any startup settings you changed in the first step.

Uninstalling from Linux

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

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

Installing DFC

Installation requirements

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

- `/usr/dt/bin` and `/usr/openwin/bin` are on the path
- `DISPLAY` is set to `localhost:0.0`
- XWindows is installed on the Linux host to run the graphical installation program. The `xterm` program may be installed in various locations depending on the operating system and software packages installed.
- Following RPM packages are available before installing DFC on a 64-bit Linux system:
 - `libXp....i686`
 - `libXi....i686`
 - `libXtst....i686`
 - `libXt....i686`
 - `libgcc....i686`
 - `libXrender....i686`
 - `xeyes`
 - `gnome-packagekit`
- Random generator on Linux is enabled: A connection request through SSL waits for a random number to be created and a delay is noticed. To avoid the delay, you can enable or disable the random generator on the Linux machine. Perform the following boot safe (`inittab`) on all DFC machines:

Start the random generator as root:

```
/sbin/rngd -b -r /dev/urandom -o /dev/random
```

Because the installation program provides a graphical interface, you cannot use a telnet session to install DFC. Install from the system console, or use an X server to perform the installation remotely.

However, be careful when you install remotely with a DISPLAY setting to localhost:0.0, as the output will be sent to that terminal rather than the one at which you are working.

Before installing DFC on Windows system, disable the IP Helper service from the Windows Services console. This method disables the Teredo Tunneling Pseudo-Interface. Restart the machine.

Installing DFC on a Windows system

1. (Linux) Set environment variables, as described in [Establishing the environment for DFC, page 391](#) and [Table 16, page 393](#).
2. Run the installation program, dfcSetup.exe (Windows) or dfcSetup.bin (Linux).
3. View the Welcome window, and click **Next**. Accept the license agreement and click **Next**.
4. Specify the directory into which the installation program should place the DFC programs and click **Next**.

The default value for this directory is C:\Program Files\Documentum. Refer to [Defining file system locations for DFC components, page 392](#) for information about how DFC uses the value you supply.

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

5. Specify whether to install optional components for developers, and click **Next**.
Select the **Developer Documentation** checkbox to request installation of Javadocs. The installation program places Javadocs into the help/dfc subdirectory of the DFC program root directory. After installation, to view the Javadocs, open index.html from this subdirectory.
6. Specify the root directory for Documentum user information and click **Next**.
7. Provide the connection broker information and specify if you want to enable SSL certificates:
 - **Connection Broker Port:** Type the connection broker port number.
 - **Connection Broker Host:** Type the host name. You can use an IP address or a symbolic address (for example, MyHost.MyCompany.com).
 - **Use certificates:** Select if you want to enable SSL certificates.

If **Use certificates** is selected, provide the DFC trust store information:

- **TrustStore:** The location of the DFC trust store. For example, `$DOCUMENTUM\secure\dfc.keystore`
- **Password:** The password of the trust store file.

Select **Use Default Java TrustStore** if you want to use the default Java trust store specified by JVM property `javax.net.ssl.trustStore`.

Note: The default value of the `dfc.session.secure_connect_default` attribute is `try_native_first`. If you want to connect in secure mode, change the attribute value to `secure` or `try_secure_first` in `dfc.properties` file.

Click **Next**.

8. Review the summary.

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

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

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

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

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

10. Click **Finish**.

11. (Linux) The installation program replaces copies of the shared library that it finds, but other copies may exist on the machine. It is safe to replace all of them with the current version, but if you do not want to do so, you must ensure that the old version does not precede the current version in any path environment variable that the current DFC might use. If the machine has a Content Server installation, you must manually replace the DMCL shared library that is in the server's bin directory.

Installing silently

The DFC installation program provides capabilities to support installing silently, that is, invoking the installation program from a command line and giving it a configuration file that allows the installation to proceed without further interaction.

Creating the configuration file

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

```
dfcSetup.exe -r C:\myFile.properties
```

This is the command for Windows. For other operating systems, use the appropriate executable file rather than `dfcSetup.exe`.

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

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



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

Running the installation program silently

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

```
dfcSetup.exe -f C:\myFile.properties -silent
```

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

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

Installing and configuring DFC on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.
3. Provide all the required details in the `dfc_conf.conf` file. Read the description of every field and provide valid values for each parameter.
4. Run the `dfc.sh` script.
5. To verify the installation, ensure that the `dfc.properties` file at `$DOCUMENTUM/config` is updated with the correct repository details.

IPv6 support

This section discusses the IPv6 support.

Documentum client connection process (dual-stack mode)

During startup, Content Server projects its host information such as IP address, port number, and so forth to its connection broker, which maintains the list of all Content Servers to which a client can connect. The client DFC connects to Content Server using IPv6 or IPv4. The client DFC first tries to communicate using IPv6. If no connection can be established with IPv6, the client DFC tries IPv4. When the client DFC exhausts both IPv6 and IPv4 connectivity options, an error message is displayed. The client DFC caches the connectivity option for the last successful connection to the Content Server. This improves performance by avoiding repeated attempts to communicate with connection options in which no connectivity can be established.

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

Configuring the client DFC

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

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

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

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

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

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

Configuring the Java Virtual Machine for IPv4 only

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

Troubleshooting installer problems

The installation program maintains an error log, which it writes to a file called `setupError.log` in the current working directory. If it cannot write into the working directory, it writes to the home directory of the user who initiated the installation.

Documentum Foundation Services

This chapter describes how to deploy the EMC Documentum Foundation Services (DFS) web services to a supported web application server, as well as information about configuration of the DFS server environment.

Introduction

Documentum Foundation Services provides a set of technologies that enables service-oriented programmatic access to the EMC Documentum Content Server platform and related products. The intended audience is system administrators or programmers who must deploy DFS.

About DFS deployment

Local and remote DFS applications

One of the first considerations in planning an application using DFS is whether to use the local DFS Java services, or whether to use the remote DFS web services.

The *local* DFS services are packaged in Java libraries available in the DFS SDK. A local service consumer runs in the same Java Virtual Machine as the services that it invokes, which connect to EMC Documentum repositories using an underlying DFC client. For more information on developing local DFS services, refer to the *EMC Documentum Foundation Services Development Guide*.

The *remote* DFS services are SOAP-based web services hosted in a J2EE servlet container. In a remote DFS application, the DFS web services are invoked by a SOAP client.

Supported environments

Before you begin deployment, review the [EMC E-LAB Interoperability Navigator](#) and the product Release Notes to make sure that the environment to which you are deploying is supported. The [EMC E-LAB Interoperability Navigator](#) contains the information about the supported web application server environments. Make sure that you are using a supported web application server version on a

supported operating system, and that any required updates from the web application server vendor have been applied. Ensure that your web application server is running on a supported version of the JVM. The [EMC E-LAB Interoperability Navigator](#) contains the information.

You must install a DFS server on a machine that is different from the one on which Content Server is installed.

Clustered deployment is supported on all supported web application servers. In clustered deployments, DFS must be deployed to each node in the cluster. In addition, the class loader settings on each node must be configured according to the instructions for the specific web application server.

Web services archive files

To deploy DFS web services, you will need one of the following two files:

- emc-dfs.ear
- emc-dfs.war

The emc-dfs.war file is supported for deployment on Tomcat only. For all other supported containers, deploy emc-dfs.ear.

Clustered deployment for load balancing

Generally DFS web services can be deployed in web application server clusters. Deployment of DFS to clusters is supported only for purposes of load balancing, and requires sticky sessions.

Other than the requirement to configure sticky sessions, there is no procedure specific to DFS for clustered deployment, so you can follow the procedure recommended by the web application server vendor for deploying web applications in a clustered configuration.

Failover not supported

While DFS supports clustered deployments for purposes of load balancing, it does not support failover (high availability) “out of the box.” There may be workarounds that will enable failover using third-party products, which are likely to impose limitations on functions available to DFS client applications (such as registered service contexts, cached queries, and client-orchestrated UCF). To pursue these options you should contact your account manager and arrange for a consultation with EMC Professional Services.

Configuration

This section covers configuration settings that are applicable to DFS deployments on any web application server.

General JVM configuration settings

To provide adequate heap space and PermGen space, we recommend the following JVM settings:

- -Xms512m
- -Xmx512m
- -XX:MaxPermSize=128m

MTOM content transfer mode settings

The DFS .NET client is based on WCF (Windows Communication Framework), which provides three modes for MTOM content transfer: buffer, streaming, and chunk. For streaming and chunk modes, WCF requires that the corresponding service operation (such as get or create) take only one argument (the input stream). This conflicts with the design of DFS, such that DFS can only use the MTOM buffer mode with a .NET client. This results in unusually high memory requirements, especially when trying to transfer large content payloads when ACS is unavailable or switched off on the server, because the entire content must be buffered in memory before transfer. Normally a .NET client will use ACS if it is available for content download operations, so under typical conditions the memory limitation is not encountered. However, if ACS content is unavailable, or if the client attempts to upload a very large content stream to the server using MTOM content transfer mode, it may result in exceeding the server's capacity to buffer the content.

Use the following configurations to avoid the memory leak issue:

- Enable ACS/BOCS for content download operations. To ensure that the `urlContent` type is returned by DFS, use the `urlReturnPolicy` setting. The client can use the `urlContent` returned by DFS to request content transfer from the ACS server.
- Use UCF as the content transfer mode for content download operations. UCF will orchestrate content transfer in both directions between the client and the ACS server.
- Optionally, make sure that both the DFS .NET client and JVM that runs the DFS server have enough memory to buffer the content. However, be aware that in this case the application will be limited to transfer of content in the range of hundreds of megabytes for a 32-bit JVM, because on most modern 32-bit Windows systems the maximum heap size will range from 1.4G to 1.6G. Although this specific limitation will not apply to 64-bit versions of Windows, the issue will still exist if you do not ensure that there is sufficient heap space to buffer very large objects in memory.

DFC configuration

The `dfc.properties` file provides property settings for the Documentum Foundation Classes runtime that DFS depends on. This file is located in `APP-INF/classes` if you are deploying the EAR file, or in `WEB-INF/classes` if you are deploying the WAR file.

If you prefer, you can use a `#include` statement to point to a properties file outside of the web application on the local file system. This can make access to some settings more convenient and allows you to modularize your configuration settings:

```
#include C:\Documentum\config\dfc.properties
```

Docbroker and global registry properties

The `dfc.properties` file includes the critical settings that are required for DFS to reach a connection broker (also called a docbroker) and connect to a Content Server.

Table 18. `dfc.properties` connect and global registry properties

Property	Value
<code>dfc.docbroker.host[0]</code>	The fully qualified hostname for the connection broker. You can add backup hosts by adding new properties and incrementing the index number within brackets.
<code>dfc.docbroker.port</code>	If you wish to use a port for the connection broker other than the default of 1489, add a port key.
<code>dfc.globalregistry.repository</code>	The global registry repository name.
<code>dfc.globalregistry.username</code>	The username of the global registry user. The global registry user, who has the default username "dm_bof_registry", must have read access to objects in the <code>/System/Modules</code> and <code>/System/NetworkLocations</code> only.
<code>dfc.globalregistry.password</code>	An encrypted password value for the global registry user.

You can either copy the username and encrypted password for the global registry user from the `dfc.properties` file on the global registry Content Server host, or you can select another global registry user and encrypt the password using the following command:

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

Properties required by the Schema service

The following two properties are required for SchemaService performance:

```
dfc.cache.ddinfo.size=10000
dfc.cache.type.currency_check_interval=86400
```

- The `dfc.cache.ddinfo.size` property can take values ranging from 1 to 10000.
- The `dfc.cache.type.currency_check_interval` property can take values ranging from 0 to 86400.

Reusing privileged DFC client instance

To reuse the privileged DFC client instance, retain the IP address or hostname of the web application server machine and specify the following in the `dfc.properties` file:

```
dfc.security.keystore.file=
#dfc.keystore location
```

Trusted login on the same host as Content Server

Trusted login is disabled by default. If required, you can set the `dfc.session.allow_trusted_login` property.

Configuring dfs-runtime.properties

DFS configuration settings are set in the `dfs-runtime.properties` file. This file is located in the `emc-dfs-rt.jar` file that resides in the `emc-dfs.ear` file. If you edit this file, repackage it in the `emc-dfs.ear` file with the same file and packaging structure. To avoid the inconvenience of unpacking the jar file to modify these properties, there are two options for using property files:

- Use an external configuration file that can be located in `APP-INF/classes` if you are deploying the EAR file, or in `WEB-INF/classes` if you are deploying the WAR file. You must rename this file `local-dfs-runtime.properties`.
- If you are using the productivity layer runtime (in a web application using DFS in local mode), you can use an external properties file on the local file system specified by the JVM `-Ddfs.runtime.properties.file` parameter.

For example:

```
-Ddfs.runtime.properties.file=
C:\\DFS\\config\\dfs-runtime.properties
```

The following list describes the precedence that these files take depending on their location:

- a `local-dfs-runtime.properties` file in the local classpath
- a runtime properties file specified with `-Ddfs.runtime.properties.file` (in productivity layer deployments only)
- a `dfs-runtime.properties` packaged with `emc-dfs-rt.jar`

The DFS application must be restarted after any changes to the configuration. As a best practice, use the provided configuration file that is deployed in the `emc-dfs-rt.jar` file for your base settings and use an external file to override settings that you specifically wish to change.

Required settings

Two DFS property settings, *tracing.enabled* and *resource.bundle*, are required. An exception is thrown if either of the setting is not provided in the configuration file. The settings in the properties file as delivered are as follows:

```
# mandatory runtime properties
tracing.enabled = false
resource.bundle = dfs-messages
resource.bundle.1 = dfs-services-messages
resource.bundle.2 = dfs-bpm-services-messages
```

Configuration settings in `dfs-runtime.properties`

By modifying the corresponding property in the `dfs-runtime.properties` file, you can perform the following tasks:

- Enable or disable AspectJ tracing for DFS.
- Specify the maximum number of elements that `QueryService` can return.
- Specify the cache size (in elements/rows) for the query.
- Specify the amount of cache to retain if a query result is not stored in the cache.
- Specify the period for the cache housekeeper to run for the Query service.
- Specify the period for the query cache manager to clean the expired cache.
- Specify the maximum number of elements in the query cache manager.
- Specify the period for the cache housekeeper to run for the Agent service.
- Set the `ContextRegistry` service expiration variables.
- Specify how often expired service contexts are being removed from the cache to free up resources.
- Specify the expiration in minutes of DFS temporary content files since the last modification date.
- Specify the initial interval of DFS scheduled task for temporary content files cleanup.
- Specify the directory in which DFS temporary content files are stored; if not specified, the temporary directory is determined.
- Specify whether to throw an exception if any downloaded JAR (like `ucf-installer.jar`) is not signed using a trusted certificate.

More detailed property descriptions are presented as comments in the `dfs-runtime.properties` file. For more information about these properties, see the `dfs-runtime.properties` packaged in `emc-dfs-rt.jar`.

Single sign-on properties

The `dfs-sso-config.properties` file specifies values that the DFS services use to process HTTP requests and pass expected values to the SSO plug-in via DFC. This file is located in `WEB-INF/classes` for the WAR package and located in `APP-INF/classes` for the EAR package. Note that this configuration file does not apply to Kerberos authentication.

This file is required in the DFS web service application. The client productivity layer runtime will also use this file to determine the header and cookie names to use for the user name and password in the HTTP request to the DFS service. If the file is not found, the DFS client runtime will use default values for the header and cookie name. Therefore, if you want to use special (non-default) names for the header and cookie in a DFS application that uses the client productivity layer, you should copy `dfs-sso-config.properties` to a directory on your DFS client application's classpath.

Table 19. dfs-sso-config.properties

Property Name	Description
sso.type	The SSO server type, supported values are "dm_rsa" and "dm_netegrity".
user.header.name, user.header.name.<integer value>	A list of possible names of HTTP headers in the HTTP request that can potentially contain user names. If more than one header from the list is found in the HTTP request, DFS uses the first header from the list that it finds.
password.cookie.name	The name of the cookie in the HTTP request that contains the SSO ticket.
sso.argument	The SSO server address. Leave this value blank for Netegrity.

Some sample settings for Netegrity Siteminder are shown below:

```
# type of single sign on server sso.type = dm_netegrity
# list of possible names of headers that contain user name.
# In case there will be more than one header with
# user name the first found header will be used.user.header.name = SM_USER
# name of the cookie containing the sso ticket password.cookie.name = SMSESSION
# sso argument to specify SSO proxy server
# so that SSO plugin in content server side will know the SSO server address
# sso.argument =
```

Configuring SSL for DFS

This section discusses how to configure secure socket layer (SSL) for DFS. Settings on various web application servers may differ from each other. For detailed operations to enable SSL for a specific web application server, see the corresponding documentation.

To configure SSL for DFS, follow these steps:

1. Get a certificate. You can purchase certificates issued by certificate authority (CA). Alternatively, you can create self-signed certificates.
2. Enable SSL on the web application server that hosts DFS. For more details, see the documentation of your web application server.
3. On the DFS client, test whether SSL is correctly enabled on the web application server by entering the following URL in a browser:

```
https://<server>:<port>/services/core/ObjectService?WSDL
```

If SSL is correctly enabled, the browser displays the content of the WSDL file.

4. Export the certificate to a local file by using the certificate export wizard for later use.
5. Import the certificate on the client:

a. DFS client:

- If you use the DFS .NET productivity layer, import the certificate to the Windows certificate manager.
- If you use the DFS Java productivity layer, import the certificate to the keystore of the local JRE that runs the DFS client.

Sample command:

```
%JRE_HOME%\bin\keytool
-import -alias test -file dfs.cert -keystore
"%JRE_HOME%\lib\security\cacerts"
-keypass <password>
```

Note: keytool is a key and certificate management utility. The sample command is for reference purpose only. For more information about the keytool utility, see Oracle documentation.

b. UCF client:

- If you use the UCF Java client and use the private JRE to launch the UCF Java client, import the certificate to the keystore of the private JRE.
- No additional configuration is needed to launch the UCF .NET client.

6. If you use the DFS .NET productivity layer, modify the application configuration file (app.config or web.config). Update the security mode attribute to "Transport" as below:

Note: If the application configuration does not exist, you have to create one. For more samples of the configuration file, see the DFS SDK.

```
<system.serviceModel>
<bindings>
  <basicHttpBinding>
    <binding name="DfsAgentService"...>
      <security mode="Transport">
        <transport clientCredentialType="None"
          proxyCredentialType="None" realm="" />
        <message clientCredentialType="UserName"
          algorithmSuite="Default" />
      </security>
    </binding>
    <binding name="DfsContextRegistryService" ...>
      <security mode="Transport">
        <transport clientCredentialType="None"
          proxyCredentialType="None" realm="" />
        <message clientCredentialType="UserName"
          algorithmSuite="Default" />
      </security>
    </binding>
    <binding name="DfsDefaultService" ...>
      <security mode="Transport">
        <transport clientCredentialType="None"
          proxyCredentialType="None" realm="" />
        <message clientCredentialType="UserName"
          algorithmSuite="Default" />
      </security>
    </binding>
  </basicHttpBinding>
</bindings>
</system.serviceModel>
```

Deployment on web application servers

The following sections provide information on deployment of DFS, or of any ECS services, to supported web application servers.

Apache Tomcat

Make sure that the Tomcat JVM settings meet the recommendations specified in [General JVM configuration settings, page 405](#). Apache Tomcat does not support EAR archives. Therefore, to deploy on Tomcat, download `emc-dfs.war` for DFS (or the provided WAR file for deploying another set of ECS services).

To perform a simple war file deployment:

1. Copy the WAR file to the `<TomcatHome>/webapps` directory.

Tomcat unpacks the WAR file to the `<TomcatHome>/webapps/<application_name>` directory, where `<application_name>` is the name of the WAR file without the file extension. The web application name is included in the service address as follows:

```
http://localhost:8080/emc-dfs/services/<module
name>/<service name>
```

In DFS productivity-layer clients, modify the `contextRoot` settings `dfs-client.xml` as required.

If you do not want to specify the web application name in the address, deploy DFS as the Tomcat ROOT application by renaming the WAR file to `ROOT.war` and deleting or renaming the `webapp/ROOT` application directory. The services are available at the location as it is shown throughout the DFS documentation:

```
http://localhost:8080/services/<module
name>/<service name>
```

For more information, refer to the Apache Tomcat web site.

VMware vFabric tc Server

You deploy the `emc-dfs.war` for DFS (or the provided WAR file for deploying another set of ECS services) to the vFabric tc Server. For more information about deploying web applications, refer to the VMware vFabric tc Server web site.

Oracle WebLogic Server

1. Set the JVM options in the WebLogic startup script.

On Windows, modify the `setDomainEnv.cmd` script; on Linux systems modify the `setDomainEnv.sh` script.

Include the following property settings in the `JAVA_OPTIONS` environment variable:

```
-Dcom.sun.xml.ws.api.streaming.XMLStreamReaderFactory.woodstox=true
```

```
-Dcom.sun.xml.ws.api.streaming.XMLStreamWriterFactory.woodstox=true
```

2. Deploy the DFS EAR file using the WebLogic Console.

For more information, refer to the Oracle WebLogic Server web site.

Note:

- When you deploy DFS on Weblogic 12c running JDK7, the location of the JAAS configuration file must be specified in WebLogic 12c by using a JVM parameter (for example, `-Djava.security.auth.login.config=<path>/jaas.config`). Specifying the location in the `web.xml` file in DFS doesn't work.
- If your deployment of DFS requires extra libraries (JAR files), follow these steps to update the `emc-dfs.ear` file to include the libraries:
 1. Copy the library into `emc-dfs.ear\lib\` directory.
 2. Specify the path of these JAR files in the `Class-Path` entry of the `MANIFEST.MF` file of the following service WARS in `emc-dfc.ear`:
 - `services-admin.war`
 - `services-bpm.war`
 - `services-ci.war`
 - `services-collaboration.war`
 - `services-core.war`
 - `services-search.war`

`MANIFEST.MF` is under the `META-INF` directory of each WAR file. Make sure that the length of each line should not exceed 70 characters in `MANIFEST.MF`.

IBM WebSphere

1. Deploy the DFS EAR file using the Integrated Solutions Console.
2. Go to **Application, Application Types, WebSphere enterprise applications**.
3. From the list of applications, select **emc-dfs**.
4. From **Detail Properties** select **Class loading and update detection**.
5. Change **Class loader order** to **Classes loaded with local class loader first (parent last)**.
6. Change **WAR class loader policy** to **Single class loader for application**.
7. Click **OK** and save changes.

If you require support for the BOCS asynchronous upload operation, then:

1. Add the new property `dfc.bof.classloader.enable_extension_loader_first=false` to the `emc-dfs.ear dfc.properties` file.
2. From the WAS admin console, go to **Server > ServerTypes > Websphere application servers**.
3. From the list of server names, select the server name in which DFS application is deployed.

4. Set the class loader policy value to **multiple**.
5. Click **Ok** and save the changes.

Deploying DFS Java productivity layer on IBM WebSphere 8.5

The DFS productivity layer contains a set of Java libraries that assist in writing DFS consumers. Using the DFS productivity layer is the easiest way to begin consuming DFS. If you need to deploy the DFS Java productivity layer client (remote mode) on IBM WebSphere 8.5, follow these steps:

1. Build the EAR or WAR file that runs the DFS productivity layer client with remote mode.
In case you use an EAR file, move the library of the root folder from .ear/APP-INF/lib to .ear/lib.
2. Move the following jar files from the EAR or WAR file you built in step one to the <WAS7_INSTALL_DIR>/AppServer/java/jre/lib/extt directory.
 - Activation.jar
 - Jaxb-api.jar
 - Jaxb-impl.jar
 - Jaxb-xjc.jar
 - Jaxb1-impl.jar
 - Jaxr-aip.jar
 - Jaxr-impl.jar
 - Jaxws-api.jar
 - Jaxws-rt.jar
 - Jaxws-tools.jar
 - Jsr173_api.jar
 - Jsr181-api.jar
 - Jsr250-api.jar
 - Mimepull.jar
 - Resolver.jar
 - Saaj-api.jar
 - Saaj-impl.jar
 - Servlet.jar
 - Sjsxp.jar
 - Stax-ex.jar
 - Streambuffer.jar
 - Xerces-impl.jar
 - Xws-security.jar

3. Restart IBM WebSphere 8.5.
4. Install the EAR or WAR file with the default configuration.
5. Start the EAR or WAR file.

If DFS SDK productivity layer and web publisher are getting installed on the same WAS instance, then follow the additional step given below:

- Set the class loader order as **PARENT_FIRST** for both DFS SDK productivity layer and web publisher applications.

If DFS SDK productivity layer and any other Documentum WDK products or DFS server are getting installed on the same WAS instance, then follow the additional step given below:

- Set the WAS server class loader policy as **Multiple**.

For more information about the DFS Java productivity layer, refer to the *EMC Documentum Foundation Services Development Guide*.

JBoss web application server

Follow these steps to deploy DFS on JBoss web application server:

1. Copy the DFS EAR file to your `<jboss-eap-6.4\standalone\deployments>` folder.
2. Restart the JBoss server.

Validating DFS deployment

To test whether your DFS deployment is successful, initially, open a service WSDL in a browser. For example, to test a DFS instance deployed on localhost at port 8080:

- Open the QueryService WSDL in your browser using the following URL:

`http://localhost:8080/services/core/QueryService?wsdl`

If you have deployed the WAR file on Tomcat and you have not deployed DFS as the Tomcat ROOT web application, then the web application name is included in the address; for example:

`http://localhost:8080/emc-dfs/services/core/QueryService?wsdl`

You can further test the deployment by running a simple sample consumer, such as the `TQueryServiceTest.java` or `QueryServiceTest.cs` consumers provided in the DFS SDK. For information about running these consumers, see the *EMC Documentum Foundation Services Development Guide*.

Deploying and configuring DFS on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.

3. Provide all the required details in the `dfs_conf.conf` file. Read the description of every field and provide valid values for each parameter.
4. Run the `dfs.sh` script.
5. To verify the installation, check `http://<dockerbaseip>:8080/services/core/SchemaService`. Also, check the web application server logs. For example, the logs at `/opt/tomcat/logs`.

Enabling Kerberos authentication

Overview

EMC Documentum supports Kerberos secure Single-Sign-On (SSO) using Microsoft Active Server Domain Services for Kerberos Key Distribution Center (KDC) services in the following ways:

- In a single domain.
- In two-way trusts between multiple domains in the same forest only; that is, cross-forest trusts are not supported.

Note: In addition, the DFS client and server must be in the same domain, whereas Content Server can be in a different domain.

You can configure DFS web services to use server-side JAX-WS handlers that interface with the Content Server Kerberos implementation. The DFS SDK includes classes that support Kerberos authentication for local Java clients, remote Java clients, and .NET clients. DFS SOAP clients that do not use the support classes in the SDK can authenticate using WS-Security headers that comply with the Kerberos Token Profile 1.1 specification.

Procedure to enable Kerberos SSO

Make sure that you have configured the following components:

- (Required for cross-domain support only) Two-way trusts between all applicable domains in the same forest
- Kerberos SSO on Content Server

Note: For more information, see the *EMC Documentum Content Server Administration and Configuration Guide*.

1. Register the DFS server's service principal name (SPN) in the Active Directory and generate a keytab file. See [Configuring the DFS server's service principal name and *.keytab file, page 416](#).
2. Enable the web application server for Kerberos. See [Configuring the DFS application server for Kerberos, page 417](#).
3. Enable Kerberos for DFS remote services. See [Enabling Kerberos for DFS remote services, page 423](#).

Configuring the DFS server's service principal name and *.keytab file

To enable authentication of the DFS server on the Kerberos Key Distribution Center (KDC), register the DFS server's service principal name (SPN) on the Active Server KDC using the Microsoft `ktpass` utility. A Kerberos SPN uniquely identifies a service that uses Kerberos authentication. In this case, the service is your DFS server. Executing the `ktpass` utility also generates a `*.keytab` file. The `*.keytab` file contains name/value pairs consisting of an SPN and a long-term key derived from a password. Both the DFS server and the KDC must be able to access the `*.keytab` file. You copy the `*.keytab` file to the DFS server machine (the machine where the Kerberos service ticket (ST) is validated) and specify the location of the `*.keytab` file in the JAAS configuration.

Note: Although the `*.keytab` file is usually used on non-Windows machines, DFS leverages the `*.keytab` file to improve network performance by eliminating Kerberos authentication communication between Windows machines and the KDC.

In some cases, you can register the SPNs of more than one DFS server to the same account. For example, in load-balanced environments support for Kerberos can be achieved by joining all load-balanced nodes into a single account and assigning a single SPN to the cluster. If access to the service is required through a different SPN (for example, based on the service host IP address rather than the load balancer name), then this SPN can also be registered with the same account. The following procedure describes the main steps for registering an SPN using a one-to-one mapping between the DFS server's SPN and user account, or a many-to-one mapping in which multiple SPNs are registered to one user account.

To configure the SPN and keytab file (main steps):

1. Create a new user (or use an existing one) for the DFS server in the Active Directory.

Note: Make sure to enable delegation trust for the service accounts who create the SPNs.

2. Map the DFS server's SPN to a user and generate the `*.keytab` file. See [Mapping the SPN to a user name, page 416](#).

Mapping the SPN to a user name

The recommended SPN format for DFS core services is:

```
DFS/<host>:<port>@<REALM>
```

EMC recommends using a host name rather than an IP address as the host string. For example, `myhost.mydomain.com`. `realm` is the name of the Kerberos realm, which is defined in the Kerberos configuration file (see [Kerberos configuration, page 418](#)).

Unless it is necessary to include a service name in the SPN, the same format is recommended for custom DFS services as well. This approach means that less configuration is required for the Active Directory because separate user accounts are not needed for every module or service. It also means that you can access all services on a `host:port` using the same service ticket (ST). It is also consistent with the way the SPNEGO protocol builds a service principal name (SPN) for a web application.

To map the SPN to a user name:

Note:

- By default, Windows Server 2008 R2 SP1 does not support DES-related ciphers (for example, DES-CBC-MD5). For more information about DES-related ciphers on Windows Server 2008, refer to the Microsoft web site.
- For the ktpass utility syntax, refer to the Microsoft web site.

1. Perform one of the following tasks:

- To map the SPN to a user name using a one-to-one mapping, execute the ktpass utility as follows:

Note: For a one-to-one mapping, do not map the same SPN to more than one user account.

```
ktpass /pass <password>
-out <keytab_file> -princ <SPN>
-crypto <crypto_type> +DumpSalt -ptype
KRB5_NT_PRINCIPAL +desOnly
```

```
/mapOp set /mapUser <user_name> /target
<domain_controller>
```

- To map multiple SPNs to a user name using many-to-one mapping, perform the following steps:

a. Execute the ktpass utility as follows:

```
ktpass /pass <password>
-out <keytab_file> -princ <SPN>
-crypto <crypto_type> +DumpSalt -ptype
KRB5_NT_PRINCIPAL +desOnly /mapOp set /mapUser
<user_name> /target <domain_controller>
```

Remember the salt string and the key version number (vno) because you need to use them in step c.

b. To map the next SPN to the same user account, execute the setspn utility as follows:

```
setspn -A <SPN> <user_name>
```

c. Execute ktpass utility for the second SPN without setting with the same user as follows:

Note: Use the salt and key version number (kvno) that were displayed as the output in step a.

```
ktpass /pass <password>
-out <keytab_file> -princ <SPN> -crypto
<crypto_type> +DumpSalt -ptype KRB5_NT_PRINCIPAL
+desOnly /mapOp set +RawSalt <salt> -in
<keytab_file> -kvno <vno>
```

d. Repeat Steps b and c for each additional SPN.

Configuring the DFS application server for Kerberos

To enable Kerberos support on the DFS application server, configure the following files:

- [Kerberos configuration, page 418](#)
- [JAAS configuration, page 418](#)

Kerberos configuration

The Kerberos configuration file (typically named `krb5.ini`) specifies Kerberos settings such as the KDC address and default realm name. The realm name includes the KDC and administration server addresses.

Create the Kerberos configuration as follows and specify its location in the `web.xml` deployment descriptors (or in an expected location, as described in the sample `web.xml` under [Enabling Kerberos for DFS remote services, page 423](#)):

```
[libdefaults]
default_realm = <REALM>
forwardable = true
ticket_lifetime = 24h
clockskew = 72000
default_tkt_enctypes =
default_tgs_enctypes =
[realms]
<REALM> =
{ kdc = <kdc_server_ip>
  admin_server = <admin_server_ip> }
[domain_realm]
<domain> = <REALM>
[logging]
default = c:\kdc.log kdc = c:\kdc.log
[appdefaults]
autologin = true
forward = true
forwardable = true encrypt = true
```

<kdc_server_ip>	The IP address of the KDC server.
<admin_server_ip>	The IP address of the Administration server.
<domain>	The domain in which the DFS server's SPN exists.
<REALM>	The realm name. For example: SRV01.COM

JAAS configuration

The JAAS configuration file entry contains JAAS specific settings such as the `<LoginContext>` name (which is also the name of the configuration entry), settings for the Kerberos login module, the DFS server's SPN, and the location of the `*.keytab` file.

The location and format of the JAAS configuration settings might be different for each web application server. Unless otherwise specified in the web application server deployment instructions, a configuration file setting can be specified as follows:

- In a JVM command-line parameter; for example:
`-Djava.security.auth.login.config=<path_to_JAAS.config>`
- In a `<env-entry>` in `web.xml` as described in [Enabling Kerberos for DFS remote services, page 423](#).

Example 7-1. Single-Domain JAAS Configuration referring to SUN JDK

```
{
  com.sun.security.auth.module.Krb5LoginModule required
  debug=false
  principal=<SPN>
  refreshKrb5Config=true
  useKeyTab=true
  storeKey=true
  doNotPrompt=true
  useTicketCache=false
  isInitiator=false
  keyTab=<dfsuser_keytab_path>;
};
```

Example 7-2. Single-Domain JAAS Configuration referring to IBM JDK

```
{
  com.ibm.security.auth.module.Krb5LoginModule required
  debug=false
  credsType="both"
  useKeytab=<dfsuser_keytab_path>
  principal=<SPN>;
};
```

Example 7-3. JAAS Configuration referring to QUEST Libraries which support both Single Domain and Multi Domain

```
{
  com.dstc.security.kerberos.jaas.KerberosLoginModule required
  debug=false
  principal=<SPN>
  realm="DFSKDC.IIG.EMC.COM"
  refreshKrb5Config=true
  noTGT=true
  useKeyTab=true
  storeKey=true
  doNotPrompt=true
  useTicketCache=false
  isInitiator=false
  keyTab=<dfsuser_keytab_path>;
};
```

Note: In WebSphere Application Server, the JAAS configuration must be specified in
 <WAS_Installation_path>\AppServer\profiles\<APP_SERVER_NODE_NAME>
 \properties\wsjaas.conf.

<loginContext>	<p>Corresponds to the Documentum DFS web application's SPN. You replace separator characters with hyphen characters and omit the @REALM segment in the SPN. For example, the following LoginContext is derived from the corresponding SPN:</p> <ul style="list-style-type: none"> LoginContext: <pre>HTTP-myhost-mydomain-com-8080</pre> SPN: <pre>HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM</pre> <p>Note: Make sure that the SPN in the JAAS configuration matches the SPN defined in <code>dfs-runtime.properties</code>.</p>		
<LoginModule>	<table border="1"> <tr> <td data-bbox="480 678 935 1129"> <p>Specify the Kerberos login module to be used to perform user authentication.</p> <p>Single Domain:</p> <ul style="list-style-type: none"> Referring to Sun JDK: <pre>com.sun.security.auth.module.Krb5LoginModule</pre> Referring to IBM JDK: <pre>com.ibm.security.auth.module.Krb5LoginModule</pre> Referring to Quest Libraries: <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre> </td><td data-bbox="935 678 1393 1129"> <p>Multi-Domain:</p> <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre> </td></tr> </table> <p>Note: For Quest login modules, if you want to enable ticket cache, perform one of the following operations. Otherwise, disable ticket cache by setting <code>useTicketCache</code> to <code>false</code>.</p> <ul style="list-style-type: none"> Enable createTicketCache: <pre>useTicketCache=true createTicketCache=true</pre> Enable createTicketCache and specify a cache path: <pre>useTicketCache=true createTicketCache=true ticketCache=<cache_path></pre> 	<p>Specify the Kerberos login module to be used to perform user authentication.</p> <p>Single Domain:</p> <ul style="list-style-type: none"> Referring to Sun JDK: <pre>com.sun.security.auth.module.Krb5LoginModule</pre> Referring to IBM JDK: <pre>com.ibm.security.auth.module.Krb5LoginModule</pre> Referring to Quest Libraries: <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre> 	<p>Multi-Domain:</p> <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre>
<p>Specify the Kerberos login module to be used to perform user authentication.</p> <p>Single Domain:</p> <ul style="list-style-type: none"> Referring to Sun JDK: <pre>com.sun.security.auth.module.Krb5LoginModule</pre> Referring to IBM JDK: <pre>com.ibm.security.auth.module.Krb5LoginModule</pre> Referring to Quest Libraries: <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre> 	<p>Multi-Domain:</p> <pre>com.dstc.security.kerberos.jaas.KerberosLoginModule</pre>		

| <SPN> | The Documentum DFS web application's SPN. For example, for SUN and IBM login modules: ``` HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM ``` For QUEST login modules, the SPN does not contain the @ character and the string after that. For example: ``` HTTP/myhost.mydomain.com:8080 ``` |

<code><REALM></code>	(Multi-domain support only) The realm name. For example: @MYDOMAIN.MYCORP.COM
<code><dfsuser_keytab_path></code>	The path to the user account's *.keytab file on the Documentum DFS web application. For example: c:\dfsuser.keytab

JBoss 6.4: In the Jboss 6.4 Application Server, the JAAS configuration must be specified in the `standalone.xml` file, which is located in the following directory:
jboss-eap-6.4\standalone\configuration\.

In your `web.xml` file, you must set the `jaas.config` entry so that it points to the above `standalone.xml` file.

Jboss 6.4 uses JAAS configurations that are included in its `standalone.xml` file. Any configuration settings that are made to any other files are ignored by the Jboss 6.4 Application Server.

Element Id	Description
<code><security-domain></code>	<p>Corresponds to the Documentum DFS web application's SPN. You replace separator characters with hyphen characters and omit the @REALM segment in the SPN.</p> <p>For example, the following Security domain is derived from the corresponding SPN:</p> <ul style="list-style-type: none"> • security-domain: HTTP-myhost-mydomain-com-8080 • SPN: HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM <p>Note: Make sure that the SPN in the JAAS configuration matches the SPN defined in <code>dfs-runtime.properties</code>.</p>

Element Id	Description
<LoginModule>	<p>Specify the Kerberos login module to be used to perform user authentication.</p> <p>For Single Domain:</p> <ul style="list-style-type: none"> • Sun JDK: <code>com.sun.security.auth.module.Krb5LoginModule</code> • IBM JDK: <code>com.ibm.security.auth.module.Krb5LoginModule</code> • Quest Libraries: <code>com.dstc.security.kerberos.jaas.KerberosLoginModule</code> <p>For Multi-Domain:</p> <ul style="list-style-type: none"> • <code>com.dstc.security.kerberos.jaas.KerberosLoginModule</code> <p>Note: For Quest login modules, when you want to enable ticket cache, perform one of the following operations:</p> <ul style="list-style-type: none"> • Enable createTicketCache: <pre>useTicketCache=true createTicketCache=true</pre> • Enable createTicketCache and specify a cache path: <pre>useTicketCache=true createTicketCache=true ticketCache=<cache_path> 32</pre> <p>Otherwise, disable ticket cache by setting <code>useTicketCache</code> to <code>false</code>.</p>
<SPN>	<p>The Documentum DFS web application's SPN. For example, for SUN and IBM login modules: <code>HTTP/myhost.mydomain.com:8080@MYDOMAIN.MYCORP.COM</code></p> <p>For QUEST login modules, the SPN does not contain the @ character and the string after that. For example: <code>HTTP/myhost.mydomain.com:8080</code></p>
<REALM>	<p>(Multi-domain support only) The realm name. For example: <code>@MYDOMAIN.MYCORP.COM</code></p>
<dfsuser_keytab_path>>	<p>The path to the user account's *.keytab file on the Documentum DFS web application. For example: <code>C:\dfsuser.keytab</code></p>

Example 7-4. The standalone.xml File Entry

```

<security-domain name="HTTP-myhost-mydomain-com-8080" cache-type="default">
  <authentication>
    <login-module code="com.dstc.security.kerberos.jaas.KerberosLoginModule"
      flag="required">
      <module-option name="storeKey" value="true"/>
      <module-option name="useKeyTab" value="true"/>
      <module-option name="principal" value=" HTTP/myhost.mydomain.com:8080"/>
      <module-option name="keyTab" value="C:/kerberos/dfs.keytab"/>
      <module-option name="doNotPrompt" value="true"/>
      <module-option name="debug" value="true"/>
      <module-option name="useTicketCache" value="false"/>
      <!--<module-option name="refreshKrb5Config" value="true"/> -->
      <module-option name="noGTGT" value="true"/>
      <module-option name="realm" value=" MYDOMAIN.MYCORP.COM "/>
    </login-module>
  </authentication>
</security-domain>

```

Enabling Kerberos for DFS remote services

To enable Kerberos authentication in remote DFS services, you must add specific libraries to the DFS archive prior to deployment and configure server-side JAX-WS handlers in deployment descriptors. This procedure applies to both the core services delivered with the DFS product as well as to custom services that are upgraded to the DFS 6.6 (or higher) runtime.

To enable Kerberos support in DFS remote services:

1. In the DFS remote services's EAR file's APP-INF/classes/authorized-service-handler-chain.xml file or WAR file's WEB-INF/classes/authorized-service-handler-chain.xml file, insert a descriptor for the Kerberos Token Profile 1.1 Support handler *before the Context Local Registry handler* as follows:

```

<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>Authorization</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.handler.AuthorizationHandler
      </handler-class>
    </handler>
    <handler>
      <handler-name>Kerberos Token Profile 1.1 Support</handler-name>
      <handler-class>com.emc.documentum.fs.rt.handlers.
        KerberosTokenServerHandler</handler-class>
    </handler>
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-class>
        com.emc.documentum.fs.rt.impl.handler.ServerContextHandler
      </handler-class>
    </handler>
  </handler-chain>
</handler-chains>

```

2. In the web.xml deployment descriptor, specify <env-entry> settings (which can be inserted anywhere inside the <web-app> element).

For EAR files, modify the `web.xml` for each module (for example, `emc-dfs.ear/services-core.war/WEB-INF/web.xml`). The DFS services' SPN name setting is required.

- For single-domain support, specify an `<env-entry>` element for each of these configuration properties:

- DFS services' SPN:

```
DFS/<HOSTNAME>:<PORT>@<REALM>
```

- `jaas.config`

- `krb5.config`

For example:

```
<env-entry>
  <description>Mandatory property defining the SPN of the DFS module
  serviced by the handler. The SPN is defined
  at deployment time, when the KDC is configured.
  The KDC realm is required as part of the SPN.</description>
  <env-entry-name>dfs.spn</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>DFS/myhost.mydomain.com:8080@SRV01.COM</env-entry-value>
</env-entry>
<env-entry>
  <description>Optional, path and name of the JAAS config
  file. If not specified here, this location can be set in a JVM command
  line parameter:
  -Djava.security.auth.login.config=/path/to/JAAS.config
  </description>
  <env-entry-name>jaas.config</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>c:/krbClient.conf</env-entry-value>
</env-entry>
<env-entry>
  <description>Optional, path and name of the Kerberos
  config file. By default, the login module
  tries to locate it in
  1. The file referenced by the Java property java.security.krb5.conf
  2. $java.home/lib/security/krb5.conf
  3. c:\winnt\krb5.ini on Microsoft Windows platforms
  4. /etc/krb5.conf on Linux platforms.
  </description>
  <env-entry-name>krb5.config</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>c:/winnt/krb5.ini</env-entry-value>
</env-entry>
```

- For multi-domain support, specify an `<env-entry>` element for each of these configuration properties:

- DFS services' SPN:

```
DFS/<HOSTNAME>:<PORT>
```

This SPN must not contain the realm (for example, @DFSSERVER.IIG.EMC.COM).

- `jaas.config`:

For example:

```
C:/jaas.conf
```

- `krb5.config`:

For example:

C:/Windows/krb5.ini

```
— jcsi.nameservers:
  <name_server_ips>
```

Note: This setting is required for multi-domain support.

For example:

```
<env-entry>
  <description>Required for Kerberos Multi-Domain, IP addresses for Kerberos
  name servers. If not specified here, this location can be set in a JVM
  command line parameter:
  -Djcsi.kerberos.nameservers=ip_addresses
  </description>
  <env-entry-name>jcsi.nameservers</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>137.10.69.244</env-entry-value>
</env-entry>
```

```
— jcsi.maxpacketize:
```

For example:

```
<env-entry>
  <description>The maximum packet size setting for multi-domain
  Kerberos support. Quest libraries use
  TCP as default protocol over UDP for communicating
  with KDC. It uses Nagle's algorithm when
  Kerberos requests are small (less than
  an Ethernet packet size, e.g. 1420) and that
  causes a delay. Quest still supports
  UDP if the customer wants to use this protocol.
  Switching from TCP to UDP can be done
  by setting the following property. If
  the packet size is less than or equal to the
  value provided to this property then
  the Quest library uses UDP to communicate
  with KDC; otherwise it uses TCP. Alternatively,
  the value can be set in a JVM command
  line parameter: -Djcsi.kerberos.maxpacketize=0
  This parameter is required for multi-domain
  support only. The default is not-set. </description>
  <env-entry-name>jcsi.maxpacketize</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>0</env-entry-value>
</env-entry>
```

3. Add the following files from the SDK to APP-INF/lib (in the EAR) or WEB-INF/lib (in the WAR):

```
commons-codec-1.3.jar (only if it does not already exist in the archive)
commons-lang-2.4.jar (only if it does not already exist in the archive)
krbutil.jar
jcifs-krb5-1.3.1.jar
vsj-license.jar (required for multi-domain support)
vsj-standard-3.3.jar (required for multi-domain support)
questFixForJDK7.jar (required for multi-domain support on JDK 7)
```

4. Repackage and redeploy the archive.
5. Create a krb5.ini file in the location specified in web.xml. See [Kerberos configuration, page 418](#).

Kerberos Diagnostics

The DFS Kerberos token server handler provides the Kerberos authentication with specific logging information. To debug the Kerberos authentication on the DFS server side, set the log level of the `com.emc.documentum.fs.rt.handlers` package to `DEBUG` in the `log4j.properties` file.

JAAS/GSS-API also provides options to enable logging for the Kerberos authentication. For more information on this and error codes for troubleshooting, refer to the Oracle web site.

Enabling SAML authentication in DFS services

DFS server-side changes

In the DFS remote services's EAR file's `APP-INF/classes/authorized-service-handler-chain.xml` file or WAR file's `WEB-INF/classes/authorized-service-handler-chain.xml` file, insert a descriptor for the SAML Profile 2.0 Support handler *before the Context Local Registry handler* as follows:

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler><handler-name>Authorization</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.handler.AuthorizationHandler
    </handler-class>
    </handler>
    <handler>
      <handler-name>SAML</handler-name>
      <handler-class>com.emc.documentum.fs.rt.handlers.SamlAssertionServerHandler
    </handler-class>
    </handler>
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler
    </handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

DFS client-side changes that are using DFS SDK APIs (productivity layer)

Perform one of the following methods on the DFS client-side to enable SAML authentication:

1. Invoke the Assertion Consumer Services URL (`http://adfs.oauth.server/adfs/ls/`) and get the SAML authentication response. Then, send the SAML response as

"dm_saml=<SAML_AUTH_RESPONSE>" in the password field. [Method 1, page 427](#) contains the information.

2. Invoke the SAML SOAP Endpoint and get the digitally signed SAML assertion response. Encode the assertion response for BASE64 and send it as "dm_saml=<SAML_AUTH_RESPONSE>" in the password field. [Method 2, page 427](#) contains the information.

Method 1

1. DFS productivity layer clients should form the authentication request for SAML server. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:AuthnRequest
AssertionConsumerServiceURL="https://localhost:8443/SAML/CreateRequest"
ForceAuthn="false" ID="<Unique ID>" IsPassive="false"
IssueInstant="2016-02-24T13:36:07.026Z"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Version="2.0" xml:space="preserve"
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
https://adfs.oauth.server
</saml:Issuer>
</saml2p:AuthnRequest>
```

The *SAML documentation* contains more details on the SAML request.

2. The authentication request XML in [Step 1](#) is encoded for BASE64 and appended to the Assertion Consumer Services URL. For example:

```
http://adfs.oauth.server/adfs/ls/?SAMLRequest=<BASE64_ENCODED_REQUEST>
```

3. You get the encoded SAML response provided by ADFS to the client. For example:

```
PHNhbWxwOlJlc3Bvb3N1IEIlePSJfOWExNGY5NWQtYzBkNC00ZDM4LTkz
NWYtMDBlMTBhOTAwMjQzIiBZWXXJzaW9uPSIyLjAiIElzc3VlSW5zdGFudD0iMjAxNi0w
Mi0yMlQwODoyMzo.....L3RydXN0PC9Jc3N1ZXI+PHNhbWxwOlJlc3Bvb3N1Pg==
```

4. Form the password for the repository user in a valid format. For example:

```
"dm_saml=<SAML authentication response received in Step 3>"
```

5. Invoke the DFS service.

Method 2

DFS client invokes the Mex Endpoint (by default, it is anonymous) for consumption on ADFS2 (if not disabled) at <https://youradfsserver.com.au/adfs/services/trust/mex>. You can use a tool like SoapUI and consume the Mex Endpoint as the WSDL. Then construct the SOAP messages and invoke the SOAP Endpoint to get the valid assertion.

The SOAP Endpoint for requesting SAML token using WS-Trust 1.3 and mixed security would be <https://youradfsserver.com.au/adfs/services/trust/13/usernamemixed>.

1. You can use curl API to communicate with server to get the SAML response. For example:

```
curl --insecure https://samlserver.emc.com/adfs/services/trust/13/usernamemixed
--data @saml-soap-request.xml -H "Content-Type: application/soap+xml"
```

```
--verbose -o "saml-soap-response.xml"
```

See the example [saml-soap-request.xml](#), page 428.

2. Using the example command in [Step 1](#), you get a SOAP response containing only the assertion part (with digital signature).

See the example [saml-soap-response.xml](#), page 429.

3. Extract the SAML assertion from the SOAP response message received from [Step 2](#).

4. Encode the assertion for BASE64 by using the SAML API. For example:

```
org.opensaml.xml.util.Base64.encodeBytes(assertionStr.getBytes(),  
org.opensaml.xml.util.Base64.DONT_BREAK_LINES);
```

5. Form the password for the repository user in a valid format. For example:

```
"dm_saml=<BASE64 encoded SAML assertion from Step 4>"
```

6. Invoke the DFS service.

DFS client-side changes that are not using DFS SDK APIs

If DFS DFS client invokes the Mex Endpoint (by default, it is anonymous) for consumption on ADFS2 (if not disabled) at <https://youradfsserver.com.au/adfs/services/trust/mex>. You can use a tool like SoapUI first and consume the Mex Endpoint as the WSDL. You can then construct SOAP messages and invoke the SOAP Endpoint to get the valid assertion.

The SOAP Endpoint for requesting SAML token using WS-Trust 1.3 and mixed security is:

<https://youradfsserver.com.au/adfs/services/trust/13/usernamemixed>

1. You can use curl API to communicate with server to get the SAML response. For example:

```
curl --insecure https://samlserver.emc.com/adfs/services/trust/13/usernamemixed  
--data @saml-soap-request.xml -H "Content-Type: application/soap+xml"  
--verbose -o "saml-soap-response.xml"
```

See the example [saml-soap-request.xml](#), page 428.

2. Using the example command in [Step 1](#), you get a SOAP response containing only the assertion part (with digital signature).

See the example [saml-soap-response.xml](#), page 429.

3. Extract the SAML assertion from the SOAP response message received from [Step 2](#) and embed the assertion element from [Step 3](#) in the SOAP header with the inclusion of Security element and create the DFS SOAP message.

See the example [DFS SOAP message](#), page 429.

Example 7-5. saml-soap-request.xml

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"  
  xmlns:a="http://www.w3.org/2005/08/addressing"  
  xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401  
    -wss-wssecurity-utility-1.0.xsd">  
  <s:Header>  
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/  
      ws-trust/200512/RST/Issue</a:Action>
```

```

<a:To s:mustUnderstand="1">https://samlserver.emc.com/adfs/services/
trust/13/usernamemixed</a:To>
<o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  <o:UsernameToken u:Id="uuid-6a13a244-dac6-42c1-84c5-cbb345b0c4c4-1">
    <o:Username>samlserver.emc.com\jegan</o:Username>
    <o:Password>Password@123</o:Password>
  </o:UsernameToken>
</o:Security>
</s:Header>
<s:Body>
  <trust:RequestSecurityToken xmlns:trust=
"http://docs.oasis-open.org/ws-sx/ws-trust/200512">
    <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <a:EndpointReference>
        <a:Address>urn:samlserver.emc.com</a:Address>
      </a:EndpointReference>
    </wsp:AppliesTo>
    <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/
200512/Bearer</trust:KeyType>
    <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/
200512/Issue</trust:RequestType>
    <trust:TokenType>urn:oasis:names:tc:SAML:2.0:assertion</trust:TokenType>
  </trust:RequestSecurityToken>
</s:Body>
</s:Envelope>

```

Example 7-6. saml-soap-response.xml

```

<S12:Envelope xmlns:S12="...">
  <S12:Header>
    <wsse:Security xmlns:wsse="...">
      <saml:Assertion xmlns:saml="...">
        AssertionID="a75adf55-01d7-40cc-929f-dbd8372ebdfc"
        IssueInstant="2003-04-17T00:46:02Z"
        Issuer="www.opensaml.org"
        MajorVersion="1"
        MinorVersion="1">
          <saml:AuthenticationStatement>
            <saml:Subject>
              <saml:NameIdentifier
                NameQualifier="www.example.com"
                Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
                uid=joe,ou=people,ou=saml-demo,o=baltimore.com
              </saml:NameIdentifier>
              <saml:SubjectConfirmation>
                <saml:ConfirmationMethod>
                  urn:oasis:names:tc:SAML:1.0:cm:bearer
                </saml:ConfirmationMethod>
              </saml:SubjectConfirmation>
            </saml:Subject>
          </saml:AuthenticationStatement>
        </saml:Assertion>
      </wsse:Security>
    </S12:Header>
    <S12:Body>
      .
      .
      .
    </S12:Body>
  </S12:Envelope>

```

Example 7-7. DFS SOAP message

```

<NS1:Envelope xmlns:NS1="http://schemas.xmlsoap.org/soap/envelope/"
  " xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance

```

```

"xmlns:dfsdmcorecontext="http://context.core.datamodel.fs.documentum.emc.com/
" xmlns:dfsdmcoreprofiles="http://profiles.core.datamodel.fs.documentum.emc.com/"
<NS1:Header xmlns:wss="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd"><wss:Security>
<Assertion ID="_7827ae41-ba82-46c8-808e-13200d6252a8"
IssueInstant="2016-08-29T09:41:33.649Z" Version="2.0"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"><Issuer>
http://samlserver.emc.com/adfs/services/trust</Issuer><ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#_7827ae41-ba82-46c8-808e-13200d6252a8">
<ds:Transforms><ds:Transform Algorithm=
"http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms><ds:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>1Du5Gs....9DBYqEs3dpI3FqioKoMFCf+A==</ds:SignatureValue>
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data>
<ds:X509Certificate>MIIC4DCC....3gv5PgX91QDD3sDA=</ds:X509Certificate>
</ds:X509Data></KeyInfo></ds:Signature><Subject>
<NameID>UserName</NameID>
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<SubjectConfirmationData NotOnOrAfter="2016-08-29T09:46:33.649Z" />
</SubjectConfirmation></Subject>
<Conditions NotBefore="2016-08-29T09:41:33.631Z"
NotOnOrAfter="2016-08-29T10:41:33.631Z"><AudienceRestriction>
<Audience>urn:samlserver.emc.com</Audience>
</AudienceRestriction></Conditions>
<AttributeStatement><Attribute Name="http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname">
<AttributeValue>UserName</AttributeValue>
</Attribute></AttributeStatement><AuthnStatement AuthnInstant=
"2016-08-29T09:41:33.601Z">
<AuthnContext><AuthnContextClassRef>
urn:oasis:names:tc:SAML:2.0:ac:classes:Password
</AuthnContextClassRef></AuthnContext>
</AuthnStatement></Assertion>
</wss:Security>
...</NS1:Header>
...<NS1:Body>
.....<NS2:get xmlns:NS2="http://core.services.fs.documentum.emc.com/">
.....<forObjects isInternal="false">
.....<NS3:Identities valueType="OBJECT_ID"
.....repositoryName="Documentum" xmlns:NS3=
....."http://core.datamodel.fs.documentum.emc.com/">
.....<NS3:ObjectId id="0901e24080016900" />
.....</NS3:Identities>
.....</forObjects>
.....<options>
.....<NS4:Profiles formatFilter="SPECIFIED" pageFilter="ANY" pageNumber="0"
.....pageModifierFilter="ANY" urlReturnPolicy="NEVER" format="pdf" xsi:type=
....."dfsdmcoreprofiles:ContentProfile" xmlns:NS4=
....."http://core.datamodel.fs.documentum.emc.com/" />
.....</options>
.....</NS2:get>
...</NS1:Body>
</NS1:Envelope>

```

IBM Tivoli Access Manager for e-business WebSEAL integration

To integrate the DFS service application with IBM Tivoli Access Manager for e-business WebSEAL, you will need to add a custom server-side JAX-WS SOAP handler to the application lib folder, and register it in a configuration file to add it to the application handler chain. For more information on IBM Tivoli Access Manager for e-business WebSEAL integration, refer to the *EMC Documentum Foundation Services Development Guide*.

To add a custom SOAP handler to the DFS server application

1. Open up the services EAR file and locate APP-INF/classes/authorized-service-handler-chain.xml. If you are deploying a WAR file, locate WEB-INF/classes/authorized-service-handler-chain.xml.
2. Package your custom handler in a jar and add it to APP-INF/lib (in EAR files) or WEB-INF/lib (in WAR files).
3. Insert a descriptor for your custom handler, as shown below, then save the file. Note that the handler should be inserted in the middle of the handler chain, as specified in the comments below:

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>Authorization</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.
        handler.AuthorizationHandler</handler-class>
    </handler>
    <handler-name>YourWebsealIvUserHandler</handler-name>
    <handler-class>com.acme.handler.YourWebsealIvUserHandler
      </handler-class>
    </handler>
    <!-- Any handler using ContextFactory, like
      KerberosTokenServerHandler or AuthorizationHandler must be inserted
      above this comment -->
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.
        handler.ServerContextHandler</handler-class>
    </handler>
    <!-- Any handler modifying DFS SOAP headers must
      come below this comment -->
  </handler-chain>
</handler-chains>
```

Implementing Custom SOAP Handler for DFS

If you want to write a new custom Authentication Handler, then you need to write a custom server-side JAX-WS SOAP handler by implementing `javax.xml.ws.handler.soap.SOAPHandler` interface. This is the JAX-WS standard way of writing the SOAP handlers. Then add the custom server-side JAX-WS SOAP handler to the application lib folder, and register it in a configuration file to add it to the application handler chain.

Configuring DFS Server to use Custom SOAP Handler

To configure a custom SOAP handler to the DFS server application:

1. Open the services EAR file and locate APP-INF/classes/authorized-service-handler-chain.xml. If you are deploying a WAR file, locate WEB-INF/classes/authorized-service-handler-chain.xml.
2. Package your custom handler in a jar and add it to APP-INF/lib (in EAR files) or WEB-INF/lib (in WAR files).
3. Insert a descriptor for your custom handler, as shown below, then save the file. Note that the handler should be inserted in the middle of the handler chain, as specified in the comments below:

```
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>Authorization</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.handler.AuthorizationHandler
      </handler-class>
    </handler>
    <handler-name>CustomAuthenticationHandler</handler-name>
    <handler-class>com.customer.handler.CustomAuthenticationHandler>
    </handler-class>
    </handler>
    <!-- Any handler using ContextFactory, like
    KerberosTokenServerHandler or AuthorizationHandler must be
    inserted above this comment -->
    <handler>
      <handler-name>Context Local Registry</handler-name>
      <handler-class>com.emc.documentum.fs.rt.impl.handler.ServerContextHandler
      </handler-class>
    </handler>
    <!-- Any handler modifying DFS SOAP headers must come
    below this comment -->
  </handler-chain>
</handler-chains>
```

Preserving JSESSIONID cookie name

In IBM Tivoli Access Manager for e-business WebSEAL junction creation, the -j option will modify the value of the path attribute of a Set-Cookie header to give identical cookies for different back-end applications. The -j junction option provides an additional feature to modify the cookie name by prepending a special string:

```
AMWEBJCT!<jct-name>!
```

For example, if a cookie named JSESSIONID arrives across a junction called /jctA, the cookie name is changed to :

```
AMWEBJCT!jctA!JSESSIONID
```

However, as a front-end application of the IBM Tivoli Access Manager for e-business WebSEAL proxy server, the DFS client depends on the JSESSIONID cookie for its operations. Therefore, the JSESSIONID cookie-renaming behavior should be disabled. There are two options for accomplishing this:

- Preserve the names of all cookies.

Prevent renaming of non-domain cookies across a specific -j junction by configuring that junction with the -n option.

- Preserve the names of specified cookies.

The name entry in the [preserve-cookie-names] stanza of the IBM Tivoli Access Manager for e-business WebSEAL configuration file allows you to list the specific cookie names that are not to be renamed by IBM Tivoli Access Manager for e-business WebSEAL. For example:

```
[preserve-cookie-names]
name = JSESSIONID
```

For further details, refer to the IBM web site.

Federated Search Services

The EMC Documentum Federated Search Services can integrate all content sources inside and outside an enterprise and enable end users to access all relevant content with a single query, no matter where the content is located and how the content is accessed.

Federated Search Services consists of the following components:

- Federated Search Services Server: The backend server that brokers user requests on various content sources
- Federated Search Adapters: Software components for querying specific content sources
- Adapter Development Kit: APIs for creating custom client applications and adapters

Federated Search Services installation

Pre-Installation Tasks

Before installing EMC Documentum Federated Search Services, make sure that your system meets the system requirements specified in the *EMC Documentum Platform and Platform Extensions Release Notes*.

Download the latest copy of the product from the the EMC Online Support website at <https://support.emc.com>.

Note: If you are installing Federated Search Services as part of the deployment of CenterStage™, select the installer provided in the CenterStage directory.

Preparing Installation Packages

Federated Search Services is packaged as follows:

- `fs2-<version_number>-win.zip`
- `fs2-<version_number>-sol.tar`
- `fs2-<version_number>-lin.tar`
- `fs2-<version_number>-aix.tar`

The following table lists the installation files contained in the installation packages for different operating systems:

Windows	Solaris	AIX	Linux
<ul style="list-style-type: none">• FS2_Adapter_Development_Kit.iam.zip• FS2_Server_SDK_win.zip• fs2Setup.exe• tanukisoftware.zip	<ul style="list-style-type: none">• FS2_Adapter_Development_Kit.iam.zip• FS2_Server_SDK_unix.zip• fs2Setup.bin• tanukisoftware.zip	<ul style="list-style-type: none">• FS2_Adapter_Development_Kit.iam.zip• FS2_Server_SDK_unix.zip• fs2Setup.bin• tanukisoftware.zip	<ul style="list-style-type: none">• FS2_Adapter_Development_Kit.iam.zip• FS2_Server_SDK_unix.zip• fs2Setup.bin• tanukisoftware.zip

Pre-Installation Tasks on UNIX and Linux

Make sure that you have write rights on UNIX and Linux before starting the installation procedure.

Confirm that you have installed the 64-bit RPM packages of the following libraries:

- `glibc-version-release.architecture` (e.g. `glibc-2.12-1.107.el6.x86_64.rpm`)
- `libXau-version-release.architecture` (e.g. `libXau-1.0.6-4.el6.x86_64.rpm`)
- `libxcb-version-release.architecture` (e.g. `libxcb-1.8.1-1.el6.x86_64.rpm`)
- `libX11-version-release.architecture` (e.g. `libX11-1.5.0-4.el6.x86_64.rpm`)
- `libXext-version-release.architecture` (e.g. `libXext-1.3.1-2.el6.x86_64.rpm`)
- `libXi-version-release.architecture` (e.g. `libXi-1.6.1-3.el6.x86_64.rpm`)
- `libXtst-version-release.architecture` (e.g. `libXtst-1.2.1-2.el6.x86_64.rpm`)

On UNIX and Linux, set the `DOCUMENTUM` environment variable to identify the directory into which you install Federated Search Services. You must set the `DOCUMENTUM` environment variable in the installation owner's environment. If you have other Documentum products installed on the system, the `DOCUMENTUM` environment variable might already exist.

Set the variable in the installation owner's `.cshrc` file (C shell) or `.profile` file (Bourne or Korn shells). Alternatively, set the variable in a file called by the `.cshrc` file or `.profile` file or in other ways permitted by UNIX or Linux.

Note: Ensure the following settings for UNIX systems:

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

Pre-Installation Tasks on Windows

Make sure that you have administration privileges on Windows before starting the installation.

Disabling User Access Control (UAC)

1. Navigate to **Control Panel > System and Security > Action Center**.
2. Click **Change User Account Control settings**.
3. Move the slider down to **Never notify**.
4. Click **OK**.
5. Restart the computer.

Installing Federated Search Services

If you are upgrading to a new version, [Upgrading Federated Search Services, page 439](#), describes the tasks you must perform before starting the installation process.

Installation Checklist

Before starting the installation, make sure that you have the system information at your disposal. The following table lists the parameters that you might need to set during the installation:

Parameters	Description	Your Notes
Destination directory	The directory for installing Federated Search Services. On Windows, the default path is C:\Documentum\. On UNIX and Linux, the default path is set by the DOCUMENTUM environment variable.	
Port range	The default port range is from 3000 to 3005. You must use five successive ports and the base port number must be greater than 1024.	
Virtual host name	A virtual host name for accessing the Federated Search Services server	
HTTP proxy settings	An HTTP proxy for connecting to the Internet. If authentication is required, you need to provide the proxy host and port, as well as the login and password.	

Parameters	Description	Your Notes
Notification settings	The mail server name and administrator email address	
Administrator credentials	The login and password for accessing the Federated Search Services Admin Center	

Running the Installation

1. Run `fs2Setup.exe` (Windows) or `fs2Setup.bin` (UNIX and Linux) to launch the Federated Search Services installer.
2. Click **Next** to read the license agreement.
3. Accept the license agreement and click **Next**.
4. Specify an installation directory for Federated Search Services and click **Next**.

Note: This step is automatically skipped in the following scenarios:

- A DFC Runtime Environment is already installed.
- You run the installer on UNIX and Linux on which the directory is determined by the `DOCUMENTUM` environment variable.

5. Select the product components you want to install, and click **Next**.

Federated Search Services is the required component and is installed by default. If you do not select the Adapter Development Kit, you can install it later by running the installer again.

6. Configure the network settings.

The default port range is from 3000 to 3005. If you want to set a custom port range, you must use five successive ports and the base port number must be greater than 1024.

You can set a virtual host name to access the Federated Search Services server. The RMI server and the Admin Center use the virtual host name to identify the host machine. A virtual host name also is useful when there are several network cards on the host machine.

Click **Next**.

7. If you use a proxy server, configure the proxy settings. Click **Next**.
8. Specify the mail server host name and the email address of the Federated Search Services server administrator.

The notification settings are required only for scheduled queries. Select **Send an e-mail at server startup** if you want to send an email to the administrator when the server starts.

Click **Next**.

9. Specify the administrator workshop configuration and click **Next**.
10. Verify the installation preview information and click **Install**.
11. Click **Done** to complete the installation.

Troubleshooting

The Federated Search Services installer logs the installation process and creates a `\logs` folder in the installation package to save log files. If you encounter errors during the installation, you can open the `install.log` file in the `\logs` folder to troubleshoot the errors.

InstallAnywhere also generates the following log files for the installations:

- `Federated_Search_Services_Install_***.log` for the Federated Search Services installation
- `FS2_Adapter_Development_Kit_Install_***.log` is for the Adapter Development Kit installation

Upgrading Federated Search Services

You must uninstall the older version of Federated Search Services before installing a new version. But before you uninstall, save your configurations.

Uninstalling Federated Search Services

On Windows

To uninstall Federated Search Services on Windows, you can use the **Uninstall a program** function in the **Control Panel**.

You can also uninstall Federated Search Services by running the `Uninstall.exe` files:

- Uninstalling both Federated Search Services and Adapter Development Kit: `<installation destination directory>\uninstall\fs2\Uninstall.exe`
- Uninstalling only Adapter Development Kit: `<installation destination directory>\uninstall\fs2_sdk\Uninstall.exe`

On UNIX and Linux

You can uninstall Federated Search Services by running the `Uninstall.bin` files:

- Uninstalling both Federated Search Services and Adapter Development Kit: `<installation destination directory>/uninstall/fs2/Uninstall.bin`
- Uninstalling only Adapter Development Kit: `<installation destination directory>/uninstall/fs2_sdk/Uninstall.bin`

Running the Installation in Silent Mode

You can install Federated Search Services in silent mode. The silent installation mode enables you to set installation configurations only once and then use the configurations to duplicate the installation on multiple machines. A silent installation reads the settings from the `silent.txt` configuration file that you create before the installation. The silent installation mode works on Windows, UNIX, and Linux systems. Use the silent installation mode only when the installation wizard does not work for you.

To install Federated Search Services in silent mode:

1. Download the installer files.
2. Create the `silent.txt` file with the content provided in .
3. Provide the values for all parameters.

The parameters you need to set correspond to the parameters listed in [Installation Checklist, page 437](#).

4. Run the following command:

```
fs2Setup.<installer file extension> -f <silent.txt file path>
```

For example, a valid command on Windows is `fs2Setup.exe -f C:\win64\silent.txt`.

Starting the Federated Search Services Server

The Federated Search Services server is a backend server that brokers user requests on various content sources.

On Windows

By default, the Federated Search Services server is installed as a Windows service in automatic startup mode. If you selected to start the Federated Search Services server manually during the installation, you can complete one of the following tasks to start the Federated Search Services server:

- To start the Federated Search Services server but not as a Windows service, run the following command:

```
<FS2 installation directory>\bin\aoServer.bat
```

- To start the Federated Search Services server as a Windows service, start the **Documentum FS2 Server** service in the Windows Computer Management interface.

- In Admin Center, **Servers operation** page, select one of the following links:
 - **Start FS2 with a graphic user interface** to start the `aOServer` command and display the Federated Search Services console
 - **Start FS2 without graphic user interface** to start or restart the `aOServer` command without the Federated Search Services console
 - **Start FS2 in service mode** to start the **Documentum FS2 Server** Windows service.

Note: If you cannot start or restart the Federated Search Services server with the links in this page, make sure that the Task Scheduler service is enabled and started. For more information about the Admin Center, refer to the *EMC Documentum Federated Search Services Administration Guide*.

Stopping the Federated Search Services Server

If you want to stop the service for maintenance or to uninstall it, perform one of the following actions:

- In Admin Center, select **Stop FS2** on the **Servers operation** page.
- Stop the **Documentum FS2 Server** in the Windows Computer Management interface.

Log Files

To review the standard output and error output of the Federated Search Services server, view the content of the following files in the `<FS2 installation directory>\www\logs` directory:

- `serverOut.log`
- `serverErr.log`

On UNIX and Linux

On UNIX and Linux platforms, run the `aOServer` command to start the Federated Search Services server:

```
<FS2 installation directory>/bin/aOServer
```

The Federated Search Services server screen console appears after you run the command. To stop all processes, select **Quit** from the **File** menu.

On UNIX systems, ensure that the number of files the server can open is not limited. On Solaris and AIX, use the following command to set this limit in the command shell you used to launch Federated Search Services:

```
%> unlimited descriptors
```

To start the server, run the following command:

```
%> fs2/bin/aOServer
```

Depending on the shell, the command may be `/fs2/bin/aOServer`.

Starting the Admin Center

The Admin Center is the administration interface for configuring adapters and monitoring the activity of the adapter backends and Federated Search Services server.

On Windows

By default, the Admin Center is installed as a Windows service in automatic startup mode. If you selected to start the Federated Search Services Windows services manually during the installation, you can complete one of the following tasks to start the Admin Center:

- Start the **Documentum FS2 AdminCenter** service in the Windows Computer Management interface.
- Run the following command:

```
<FS2 installation directory>\bin\aoAdmin.bat
```

Do not close the command window because it stops the Admin Center.

Stopping the Admin Center

If you want to stop the Admin Center service for maintenance reasons or if you want to uninstall it, use the Windows Computer Management interface.

Log Files

To review the standard output and error output of the Admin Center, view the content of the following files in the `<FS2 installation directory>\admin\logs` directory:

- `serviceErr.log`
- `serviceOut.log`

On UNIX and Linux

On UNIX and Linux platforms, run the `aoAdmin` command to launch the Admin Center

```
<FS2 installation directory>/bin/aoAdmin
```

You can then access the server using the following URL:

```
https://servername.your_company.com:<adminserverport>/AdminCenter
```

Note that `<adminserverport> = <fs2_server_port> + 3`. The URL is a secured protocol that starts with `https://` instead of `http://`.

On UNIX systems, ensure that the number of files that the server can open at any time is not limited. On Solaris and AIX, you can use the following command to set this limit in the command shell you used to launch Federated Search Services:

```
%> unlimited descriptors
```

To launch the Admin Center, use the same login ID that you used to install Federated Search Services. This is because the Federated Search Services installer sets conservative file access privileges on configuration files that protect personal data from security gaps. If you do not use the same login ID, the permissions or ownership on those files might be modified. Run the following command:

```
%> fs2/bin/aOAdmin
```

Note: Depending on the shell, the command may be `/fs2/bin/aOAdmin`.

Silent Installation Files

To make a silent installation the following configuration files are required. The files must not be used as is, review each parameter and provide the appropriate value.

Parameter values in bold are the default values. Parameter values in italics and in angle brackets `<>` are variables and must be set according to your needs and your current configuration.

Note: If no Java Virtual Machine (JVM) is available on the machine or if the JVM is obsolete, the JVM is installed. The JVM version 1.8.0_77 is used for Windows, Solaris, and Linux operating systems. The JVM version 1.8.0 SR3 is used for the AIX operating system.

silent.txt

```
INSTALLER_UI=silent

# The user directory to Federated Search Services. The default path is C:\\Documentum\\.
# On UNIX and Linux, the default path is set by the DOCUMENTUM environment variable.
COMMON.DCTM_USER_DIR=<user directory>

# Specify the Federated Search Services components to install.
ECI_SERVER.INSTALL_FS2=true
ECI_SERVER.INSTALL_SDK=true

# Specify the network configuration.
ECI_SERVER.DEFAULT_NETWORK_CONFIGURATION=true
# Specify the virtual host to access the Federated Search Services server.
ECI_SERVER.VIRTUAL_HOST_CONFIG=false
ECI_SERVER.ECI_HOSTNAME=<virtual host name>
# Specify the port range. You must use five successive ports and the base port number
# must be greater than 1024.
ECI_SERVER.DEFAULT_ECI_SERVER_BASE_PORT=3000
ECI_SERVER.ADMIN_HTTP_PORT=3002
ECI_SERVER.ADMIN_SECURE_HTTP_PORT=3003
ECI_SERVER.ADMIN_SERVER_SHUTDOWN=3004
ECI_SERVER.RMI_REGISTRY_PORT=3005

# SSL settings
ECI_SERVER.ADMIN_SSL_PROTOCOL=TLS
ECI_SERVER.ADMIN_ALGO=SunX509
# Use the following SSL settings if you use AIX.
ECI_SERVER.ADMIN_SSL_PROTOCOL=SSL
ECI_SERVER.ADMIN_ALGO=IbmX509

# Specify the HTTP proxy settings.
ECI_SERVER.PROXY_PORT=8000
ECI_SERVER.PROXY_HOST=<proxy host name>
```

```
ECI_SERVER.PROXY_LOGIN=<proxy login>
ECI_SERVER.PROXY_PASSWORD=<proxy password>
ECI_SERVER.PROXY_CONFIG=false
ECI_SERVER.AUTHENTICATION_CONFIG=false

# Specify the notification settings.
ECI_SERVER.MAIL_SERVER_HOST=mailhost
ECI_SERVER.ECI_ADMIN_EMAIL=<administrator email address>
# Specify whether to send an email to the administrator when the server starts.
ECI_SERVER.SEND_MAIL_CONFIG=false

# Specify the administrator configuration.
ECI_SERVER.ADMIN_ID=admin
ECI_SERVER.ADMIN_PASSWORD=ecis
# Specify whether to automatically upgrade adapters.
ECI_SERVER.UPDATER_IENABLED=true
# Specify whether to start the services after installation.
This setting applies to Windows only.
ECI_SERVER.SERVICES_ISSTARTED=true
# Specify whether to use the default administrator configuration.
ECI_SERVER.DEFAULT_ADMIN_CONFIGURATION=true
```

Setting up Docker on Federated Search Services server

Installing and configuring FS2 on Ubuntu Docker container

Prerequisites

1. Install the Docker engine.

The Content Server section contains the steps to install the Docker engine. For more information, see *Docker documentation*.

2. Pull the Docker image.

```
$ docker pull ubuntu
```

3. Run the container and modify the ports as per your requirement.

```
$ docker run -it -h HOSTNAME -p 3000-3005:3000-3005 --name fs2 ubuntu
```

For more information, see *Docker documentation*.

4. Copy installer packages.

Use the following command to copy the FS2 binaries into a directory (e.g. /home) in a Docker container:

```
$ docker cp <path-to-build-FS2> <container-name>:/home
```

Configuring Federated Search Services Server on Docker container

1. Set the environment variable for DOCUMENTUM inside the container:

```
docker$ export DOCUMENTUM=/root/dctm
```

2. Set the system locale to support UTF-8:

```
docker$ export LANG=en_US.UTF-8
docker$ locale-gen en_US.UTF-8
```

3. Modify the random number generator.

```
docker$ ln -sf urandom /dev/random
```

4. Create the silent install configuration file.

It is recommended to install FS2 in the silent mode. For more details about silent.ini file, see the *FS2 - Silent Installation Files* section.

Installing FS2 on Docker container

1. Start the silent installation.

Assume that the FS2 binaries are in the */home/fs2Installer* folder. Change the permissions for the setup file as follows:

```
docker$ chmod 755 /home/fs2Installer/fs2Setup.bin
```

Access the directory that contains the FS2 binaries, and execute **fs2Setup.bin** with **silent.ini**:

```
docker$ cd /home/fs2Installer
docker$ ./fs2Setup.bin -f silent.ini
```

2. Start and stop the services.

To start FS2 services, execute the following script files:

```
docker$ cd $DOCUMENTUM/fs2/bin
docker$ ./aOServer -nogui
docker$ ./aOAdmin
```

To stop FS2 services, run the scripts with following options:

```
docker$ ./aOAdmin -stop
docker$ ./aOServer -stop
```

Common notes for Docker environment

1. It is important that any other program that needs to connect to FS2 must be set up inside the same Docker network with FS2. For example, if Documentum Administrator needs to connect with FS2, it must be set up as a Docker image and start in a same Docker network with FS2.

Assume the internal IP address of Documentum Administrator is 172.17.0.2, the internal IP address of FS2 is 172.17.0.3, and users need to configure the `dfc.properties` file in Documentum Administrator with this internal IP address as follows:

```
dfc.search.external_sources.enable=true
dfc.search.external_sources.host=172.17.0.3
dfc.search.external_sources.port=3005
```

2. The configuration files of FS2 can be externalized outside the container on the host machine.

To do this, use “-v” option at run time.

Example

```
$ docker run -it -h HOSTNAME -p 3000-3005:3000-3005 \
-v /home/fs2Storage/admin:/root/dctm/fs2/admin \
-v /home/fs2Storage/www:/root/dctm/fs2/www \
-v /home/fs2Storage/docs:/root/dctm/fs2/docs \
-v /home/fs2Storage/jars:/root/dctm/fs2/lib/jars \
-v /home/fs2Storage/wrapper:/root/dctm/fs2/lib/wrapper \
--name fs2 ubuntu
```

In the example, the five directories containing the configuration files are stored in the `/home/fs2Storage/` folder.

Federated Search Services adapters installation

Overview

This section provides instructions for installing EMC Documentum Federated Search Services adapters and configuring backends. This section also provides information about common Federated Search Services attributes.

Always perform configuration tasks using the Admin Center. The *EMC Documentum Federated Search Services Administration Guide* provides more information about the Admin Center. This chapter contains references to configuration files and the possibility to edit them. Only experienced administrators can perform manual configuration for troubleshooting purposes.

This chapter is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. This person is responsible for the technical configuration of the system including the definition of backends.
- The librarian working in cooperation with the administrator organizes the backends into domains that make most sense to the end users.

This section describes how to integrate a new database or a new document repository using one of the enterprise adapters offered by EMC Documentum Federated Search Services.

For each of the adapters, there is a presentation of the specifications and their behavior, as well as a step-by-step procedure to explain how to integrate a new source that uses these adapters.

Third-party adapters can be harmful when run by the system. Check that the adapter can be trusted before installing it on the Federated Search Services server.

This section includes several references to *Xtrim* and *askOnce* files or programs. *Xtrim* is the development and runtime foundation of Federated Search Services and has been included as part of the file names in many circumstances. For purposes of understanding, you can assume that ECI Services, *askOnce*, and *Xtrim* are synonymous.

Properties and configurations that are identified as supported by or available in Webtop clients are also supported by and available in WDK-based clients. Refer to the client application documentation to make sure that there is no limitation or that the corresponding feature is exposed.

Main Concepts

This section describes the main concepts related to Federated Search Services adapters.

Adapters

An adapter links the Federated Search Services server to remote sources of information or document repositories. An adapter also manages the communication and interaction with a particular source. Adapters are delivered in the form of *adapter bundles*. For instance, an adapter developer develops an adapter bundle for the Microsoft Index Server search engine.

Adapter Bundles

Adapters are delivered in the form of adapter bundles. Adapter bundles are files with the `.zip` extension. An adapter bundle holds all the necessary files to communicate with a type of information source. For instance, an adapter developer would develop an adapter bundle for the Microsoft Index Server search engine and distribute the adapter bundle as `MSIndexSvr.zip`.

Backends

A backend is the configuration of an adapter bundle for a specific source of information available from a specific Federated Search Services server. For instance, to access a local source, you specialize an adapter bundle to include the host name of the source. A backend also allows the administrator to manage an adapter behavior by setting specific properties. For instance, to apply a new filtering option for duplicated results, you can change the `duplicateKey` property. Adapter backends are also called information sources.

Note: After installing a new adapter, restart the Documentum FS2 Server and Documentum FS2 AdminCenter Windows services before configuring the backends.

Configuring Adapter Backends

Adapter bundles produced by adapter developers have a default configuration that must be modified to work in your local environment. To do so, define and configure adapter backends. Typical configuration performed during the definition of a backend is the setting of the communication parameters to point to a different instance of the source.

Note: The properties `client.overview` and `client.resultIcon` are currently not exposed in client applications though they are still visible in the backend configuration files.

Configuring and Testing an Adapter Backend

A backend is defined by creating a file with the name pattern: `<Backend_name>.conf` and placing it in the same directory as the adapter bundle. The properties defined in the local configuration file override the default values from the bundle, creating a tailored backend fitting the needs of the local system.

Note: Any property marked as `Mandatory` must be explicitly defined either in the bundle configuration file (as packaged by the developer of the adapter) or in the backend configuration file (that you can edit yourself).

For most of the adapters presented here, there is no description of how to test them. All the adapters can be tested in the Admin Center at configuration time. They can also be manually tested using the `aOWrapperTester` command and `aOverboseWrapperTester` command. The `aOverboseWrapperTester` command returns more trace information. This mechanism is described in the *Testing communication with an adapter backend* section of the *EMC Documentum Federated Search Services Administration Guide*.

Configuring the Authentication

Several configuration properties are required to set the authentication at the backend level. The properties interact between each other. The following table summarizes the authentication scenarios you can set for WDK clients such as Webtop or CenterStage™ clients.

Note: The `loginPassword` attribute is encrypted using symmetric algorithm PBESWithHMACSHA-1 and AES-128 and is saved in the backend configuration file.

Table 20. Authentication Scenarios

Scenario	Configuration
The source is public or does not support authentication.	<code>supportsLogin=false</code>
The source supports authentication. The end users must specify their own credentials.	<code>supportsLogin=true</code> <code>authenticationMode=user</code>

Scenario	Configuration
<p>The source supports authentication.</p> <p>The end users are prompted for their own credentials. If not provided, a corporate account is used.</p>	<pre> supportsLogin=true authenticationMode= user_and_default_account loginName= <corporate_account_login_name> loginPassword= <corporate_account_login_password> </pre>
<p>The source supports authentication.</p> <p>A corporate account is used for all users. The end users are not prompted for their credentials.</p>	<pre> supportsLogin=true authenticationMode=default_account loginName= <corporate_account_login_name> loginPassword= <corporate_account_login_password> </pre>

Common Backend Configuration Properties

This section describes the properties that you can add in the configuration of your backends.

When a property is not visible in the Admin Center, you can add it as described in the following procedure:

1. In the Admin Center, go to the **Properties** page of the adapter backend that you want to configure.
2. At the bottom of the page, find two text boxes with the **Add** button to the right.
3. Enter the property name in the left text box and the property value in the right text box.
4. Click **Add**. The property is listed below with a **Delete** button next to it.
If you want to change an added property, delete the property and add a new one.
5. Repeat Step 3–4 if you want to add more properties.

Table 21. Backend configuration properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers) using the / character as the path separator. Usually it consists of <domain name>/<bundle_name>.jar, for example, core/DocumentumFC.jar.
Default value	<currentrelative_path>/<bundle_name>.jar
Mandatory	Yes (No, if the backend has the same name)
Visible on client	No

Table 22. Backend configuration properties — host

Item	Description
Description	The hostname of the source: an IP number or a valid DNS name such as acme.corp.emc.com.
Default value	None
Mandatory	Yes for HTTP adapter
Visible on client	No

Table 23. Backend configuration properties — port

Item	Description
Description	The port number of the server
Default value	80
Mandatory	Yes for HTTP adapter
Visible on client	No

Table 24. Backend configuration properties — protocol

Item	Description
Description	The protocol to use when connecting to the source: http or https. To disable the remote source authentication with https, use the value: https-no-auth. For example, protocol=https or protocol=https-no-auth.
Default value	http
Mandatory	No
Visible on client	No

Table 25. Backend configuration properties — protocolX

Item	Description
Description	The protocol to use at level X when connecting to the source: http or https. To disable the remote source authentication with https, use the value: https-no-auth. For example, protocol2=https or protocol2=https-no-auth
Default value	The value of the property protocol
Mandatory	No
Visible on client	No

Table 26. Backend configuration properties — home

Item	Description
Description	The URL of the information repository or web service, for example, home=http://www.altavista.com.

Item	Description
Default value	The default value is computed using the properties host, port, and protocol of the backend: <code><protocol>://<host>:<port></code> .
Mandatory	Not currently exposed in client applications.
Visible on client	Yes

Table 27. Backend configuration properties — action

Item	Description
Description	Relative URL address for the search interface of the target data source, for example, <code>action=/go/xrx/search.cgi</code> .
Default value	None
Mandatory	Yes for HTTP adapter
Visible on client	No

Table 28. Backend configuration properties — actionX

Item	Description
Description	Relative URL address at level X for the search interface of the target data source, for example, <code>action1=/go/xrx/search.cgi</code> .
Default value	The value of the property <code><action></code>
Mandatory	No
Visible on client	No

Table 29. Backend configuration properties — method

Item	Description
Description	The type of HTTP request supported by the action: post or get.
Default value	get
Mandatory	No
Visible on client	No

Table 30. Backend configuration properties — methodX

Item	Description
Description	The type of HTTP request supported by the action at level X: post or get, for example <code>method2=get</code> .
Default value	The value of the property <code>method</code> .
Mandatory	No
Visible on client	No

Table 31. Backend configuration properties — supportsLogin

Item	Description
Description	Set to true when the backend allows individual users to log in to perform a search on private or confidential data.
Default value	false
Mandatory	No
Visible on client	Yes

Table 32. Backend configuration properties — loginName

Item	Description
Description	<p>The default login name to use when authenticating with the source. This default login name is used for any end user that did not specify a personal one for this backend.</p> <p>Active only when the property supportsLogin is set to true.</p>
Default value	None
Mandatory	No
Visible on client	No

Table 33. Backend configuration properties — loginPassword

Item	Description
Description	<p>The default login password to use when authenticating with the source. This default login password is used for any end user that did not specify a personal one for this backend.</p> <p>Active only when the property supportsLogin is set to true.</p> <p>The loginPassword attribute is encrypted using symmetric algorithm PBESWithHMACSHA-1 and AES-128 and is saved in the backend configuration file.</p>
Default value	None
Mandatory	No
Visible on client	No

Table 34. Backend configuration properties — proxySet

Item	Description
Description	Set to true to use the default HTTP proxy to access to source. You can configure the HTTP proxy in <code>server.conf</code> . Refer to the <i>EMC Documentum Federated Search Services Administration Guide</i> for more information.
Default value	true

Item	Description
Mandatory	No
Visible on client	No

Table 35. Backend configuration properties — query

Item	Description
Description	The list of primary attributes available for querying this source
Default value	Inherited from adapter bundle
Mandatory	Yes
Visible on client	Yes

Table 36. Backend configuration properties — result

Item	Description
Description	The list of known attributes returned by this source
Default value	Inherited from adapter bundle
Mandatory	Yes
Visible on client	Yes

Table 37. Backend configuration properties — filter

Item	Description
Description	Set to true to post-filter the results before sending them to the client (true/false).
Default value	true
Mandatory	No
Visible on client	No

Table 38. Backend configuration properties — stopLimit

Item	Description
Description	Maximum number of results returned by an adapter, after FS2 filtering.
Default value	50
Mandatory	No
Visible on client	No

Table 39. Backend configuration properties — compoundScore

Item	Description
Description	Set to true to re-calculate score and re-rank the results based on information from the source and query.

Item	Description
Default value	true
Mandatory	No
Visible on client	No

Table 40. Backend configuration properties — expirationTime

Item	Description
Description	The expiration time of the adapter until it aborts its execution. This time is expressed in seconds. The timer is started when the server tries to establish the connection to the source. Related property in <code>server.conf</code> : <code>xtrim.maxActionFailure</code>
Default value	180
Mandatory	No
Visible on client	No

Table 41. Backend configuration properties — dateFormat

Item	Description
Description	The format of the date on the source that is used to convert dates from the FS2 date format to the format of the source, for example, MM/DD/YYYY.
Default value	YYYY-MM-DD
Mandatory	No
Visible on client	No

Table 42. Backend configuration properties — duplicate

Item	Description
Description	Set to true to remove duplicate results. Results are duplicate if they have the same value for the attribute defined as the <code>duplicateKey</code> .
Default value	true
Mandatory	No
Visible on client	Yes

Table 43. Backend configuration properties — duplicateKey

Item	Description
Description	<p>Identifies duplicate results from a source, for example, "ISBN". By default, when this attribute is absent, results are identified by the 'URL' attribute. But some sources generate different URLs for the same result across several invocations. In this case this property enables to use another attribute instead of URL to improve the identification.</p> <p>Two results are duplicates if, and only if, the values of the attribute defined in duplicateKey are identical.</p> <p>For example, Backend1: duplicateKey=title Backend2: duplicateKey not defined. Two results extracted from Backend1 are duplicates if the values of the attribute title are identical. If a result from Backend1 is compared to a result from Backend2, then the title of the first result is compared to the URL of the second one.</p> <p>The results that are not duplicates are considered <i>new</i>.</p>
Default value	URL
Mandatory	No
Visible on client	Yes

Table 44. Backend configuration properties — modificationKey

Item	Description
Description	<p>Checks if duplicate results contain the same data by comparing required attributes.</p> <p>By default, if this property is not present in the backend, all attributes are compared. Consequently, two duplicate results are considered modified if at least one attribute is different between the two results. Use the property to compare only some attributes of the results or to compare only the date attribute or the version attribute of the results.</p> <p>Compare a list of attributes: modificationKey=attribute1, attribute2, ... If at least one value of the attributes listed is different, the two results are not identical. The most recently extracted result replaces the old one.</p> <p>Compare the date attribute: modificationKey=(date)<date attribute>, for example, modificationKey=(date)last_modified_date. The result with the more recent date is retained and additional attributes from the other results are merged, if they exist.</p> <p>Compare the version attribute: modificationKey=(version)<attribute that contains an integer>, for example, modificationKey=(version)version_number. The result with the highest version number is retained and additional attributes from the other results are merged, if they exist.</p>
Default value	All attributes

Item	Description
Mandatory	No
Visible on client	Yes

Table 45. Backend configuration properties — queryLanguage

Item	Description
Description	The language used to query the source. For example, in <code>Google-Italia.conf</code> , set <code>queryLanguage=italian</code> .
Default value	None
Mandatory	Deprecated, this was only used with ECI Multilingual Services (MLS) component which was last available in ECI Services version 6.5.
Visible on client	Yes

Table 46. Backend configuration properties — encoding

Item	Description
Description	The information source encoding format. This format is used during the recovering of the information source data in order to obtain the appropriate character stream from an byte Stream.
Default value	windows-1252
Mandatory	No
Visible on client	No

Table 47. Backend configuration properties — trusted

Item	Description
Description	The list of attributes that are trusted. It means that no filtering is done on these attributes. It is useful when a source does not return the full content of an attribute, like “body” in the case of AltaVista, but we can still “trust” the selection and filtering of results implemented by AltaVista. Note: It also applies to an attribute that is never returned by the source but still is “trusted”. For example, a source supporting a search on “keywords” but that does not return a “keywords” attribute. A search on “full-text” always matches when some attributes are configured as trusted.
Default value	None
Mandatory	No
Visible on client	Yes

Table 48. Backend configuration properties — supportsSubsumption

Item	Description
Description	Set to true to allow the translation of the FS2 query into several subqueries when the OR operator is not supported. For example, the FS2 query {title,CONTAINS, printer OR scanner} is translated into query1 {title,CONTAINS, printer} and query2 {title,CONTAINS,scanner} when the OR is not supported by the source.
Default value	true
Mandatory	No
Visible on client	Yes

Table 49. Backend configuration properties — maxSubsumedQueries

Item	Description
Description	The maximum number of subqueries allowed when supportsSubsumption is set to true. If the FS2 query is translated into more subqueries than the allowed maximum value, only the first queries are sent to the source. If m is the number of subqueries and n the number of allowed subqueries, and n is lower than m , only the n first queries are sent to the source.
Default value	5
Mandatory	No
Visible on client	Yes

Table 50. Backend configuration properties — image

Item	Description
Description	Special logo used to represent results from this backend. If the value of this property is "ACME" (for example, image=ACME), store the logo in: <code>\www\tomcat\webapps\ao\data\sources\source-acme.gif</code> . Note: Image name must always be in lowercase.
Default value	<code>\www\tomcat\webapps\ao\data\sources\source-<backend_name>.gif</code>
Mandatory	Not currently exposed in client applications.
Visible on client	Yes

Each adapter backend can define custom properties for a client interface.

These properties are necessarily prefixed by "client" and are returned as is to the client interface. The following tables describe these properties.

Table 51. Backend configuration properties — client.overview

Item	Description
Description	A comma-separated list of attributes which constitutes the overview field in the client interface.

Item	Description
Default value	abstract, body
Mandatory	Not currently exposed in client applications.
Visible on client	Yes

Table 52. Backend configuration properties — client.resultIcon

Item	Description
Description	<p>The name of the icon to display in the client interface with results coming from this backend.</p> <p>If the property is not null, icon-<i><resultIcon></i>.gif image is displayed with every result coming from the adapter (height = 25 pixels).</p>
Default value	None
Mandatory	Not currently exposed in client applications
Visible on client	Yes

Table 53. Backend configuration properties — client.dfc.types

Item	Description
Description	<p>Instructs client applications that an external source supports some doctypes or can be safely searched using the attributes of some doctypes.</p> <p>For example, set client.dfc.types=doctype1,doctype2.</p>
Default value	None
Mandatory	No
Visible on client	Yes

Table 54. Backend configuration properties — zipHTTPResponse

Item	Description
Description	Set to true to support HTTP response in compressed mode.
Default value	false
Mandatory	No
Visible on client	No

Table 55. Backend configuration properties — exposeNativeQuery

Item	Description
Description	<p>Set to true to expose the query sent to the source. It enables the Show native query feature in Webtop Search. Before activating this functionality, make sure that the query does not contain sensitive information that should not be disclosed to the end user.</p> <p>For example, <code>exposeNativeQuery1=true</code>.</p> <p>Note: This property is natively supported by JDBC and HTTP adapters, but has to be implemented for Base Adapters.</p>
Default value	false
Mandatory	No
Visible on client	Yes

Table 56. Backend configuration properties — exposeNativeQueryX

Item	Description
Description	<p>Set to true to expose the query sent to the source. It enables the Show native query feature in Webtop Search. Before activating this functionality, make sure that the query does not contain sensitive information that should not be disclosed to the end user.</p> <p>For example, if there is a login page before the search form, set <code>exposeNativeQuery2=true</code>.</p> <p>Note: This property is natively supported by JDBC and HTTP adapters, but has to be implemented for Base Adapters.</p>
Default value	false
Mandatory	No
Visible on client	Yes

Table 57. Backend configuration properties — authenticationMode

Item	Description
Description	<p>Several configurations are supported, if the adapter is configured with:</p> <ul style="list-style-type: none"> • <code>authenticationMode= default_account</code>, the end users are not asked for credential at all, the corporate account is used transparently. • <code>authenticationMode=user</code>, the end users have to enter their own credentials. • <code>authenticationMode=user_and_default_account</code>, the end users are asked for credential. However they can indicate that they want to use the corporate account.

Item	Description
Default value	If loginName property is defined: authenticationMode=default_account If loginName is not defined: authenticationMode=user
Mandatory	No
Visible on client	Yes

Common FS2 Attributes

A common adapter configuration is used to define a metadata mapping between object fields in a source and FS2 attributes. One example is the properties named *mapout.XXX* and *mapin.YYY* for the Documentum adapter.

It is important to use common FS2 attributes when possible in your adapter. In Federated Search Services, a query searches several adapters to provide its result set. Therefore, all adapters share metadata. Use common FS2 attributes when possible so that a search on *title* in FS2 is handled consistently. For example, search on source A where the metadata is named *heading* and on source B where metadata is named *object_name* is only consistent if the metadata is shared.

A comprehensive list of FS2 attributes is available from the attributes localization file. The default location of the file is `<FS2 installation directory>\www\docs\conf\attributesDescription_en.properties`.

The `<FS2 installation directory>` is the default installation directory of Federated Search Services. On Windows systems, it is `C:\Documentum\fs2`.

Table 58. Common FS2 attributes — source

Item	Description
Description	Name of the backend that returns the result.
Visible on	Webtop, CenterStage It is displayed as the result origin in Webtop (like the path of a result returned by a Documentum repository). It is displayed in the My Extra Sources filter in CenterStage.
Reserved	This attribute is created by Federated Search Services and must not be generated by the adapter.

Table 59. Common FS2 attributes — score

Item	Description
Description	Indicates the relevance of the result based on the source criteria. If the rating is closer to 100%, the result matches your query better.

Item	Description
Visible on	Webtop It is displayed as the Score column in Webtop.
Reserved	This attribute is created by Federated Search Services and must not be generated by the adapter.
Format	A percentage (without the % unit sign)

Note: The score is reserved and is not expected to be listed in the properties query no result.

Table 60. Common FS2 attributes — title

Item	Description
Description	Name assigned to the resource.
Visible on	Webtop It is displayed as object_name in Webtop.

Table 61. Common FS2 attributes — URL

Item	Description
Description	Contains a hypertext link or an internet address where the resource can be found.
Visible on	Webtop It is a link on the title in all clients result list.
Format	A valid URL syntax

Table 62. Common FS2 attributes — date

Item	Description
Description	Date on which the resource was made available.
Visible on	Webtop, CenterStage It is displayed as r_modify_date in Webtop. It is displayed as the Last changed, Last accessed, and Created dates after importing the object in CenterStage.
Format	ISO 8601 format (yyyy-MM-dd) For example, May 21st, 2004 is converted into 2004-05-21.

Table 63. Common FS2 attributes — body

Item	Description
Description	Full or partial representation of the content of the resource.
Visible on	Webtop It is displayed as summary in Webtop and as the default content of the middle column in other clients result list.

Table 64. Common FS2 attributes — abstract

Item	Description
Description	A textual description of the content of the resource.
Visible on	CenterStage It is displayed as the Subject after importing the object in CenterStage.

Table 65. Common FS2 attributes — author

Item	Description
Description	Persons or organizations primarily responsible for the intellectual content of the resource.
Visible on	Webtop It is displayed as authors in Webtop.

Table 66. Common FS2 attributes — keyword

Item	Description
Description	To search among a predefined keywords list.
Visible on	Webtop It is displayed as keywords in Webtop.

Table 67. Common FS2 attributes — format

Item	Description
Description	Physical or digital manifestation of the resource.
Visible on	Webtop and CenterStage Pro It is displayed as a_content_type in Webtop. It is displayed as Format in CenterStage.
Format	A valid MIME type syntax

Table 68. Common FS2 attributes — size

Item	Description
Description	Document size (in KB).
Visible on	Webtop It is displayed as <code>r_full_content_size</code> in Webtop.
Format	Kilobytes (just the figure, no unit)

Table 69. Common FS2 attributes — rank

Item	Description
Description	Relevance of the result according to the source criteria. If the rank is closer to 1, the result matches your query better.
Visible on	Webtop It is displayed as <code>rank</code> in Webtop.
Format	Integer

Table 70. Common FS2 attributes — site

Item	Description
Description	The place where the document has been found such as Sun.Worldwide, or <code>www.sun.com</code> .
Visible on	CenterStage It is displayed in Keywords in CenterStage.

Table 71. Common FS2 attributes — collection

Item	Description
Description	The folder or catalog that contains the resource.
Visible on	CenterStage It is displayed in Keywords in CenterStage.

Documentum Adapter

The purpose of the Documentum adapter is not only to allow an FS2 client to query a Documentum repository but also to allow the end user to visualize the documents that are present on Documentum Content Server.

Principles of the Documentum Adapter

The communication between the adapter and Documentum is performed using the Documentum DFC library. The DFC library is based on the native DMCL library. The Documentum adapter is compatible with Documentum Content Server versions 5.x, 6.x, and 7.x.

Note: The Documentum adapter does not require to install the DFC on the FS2 server. Federated Search Services comes with its own set of DFC components.

Installing the Documentum Adapter

The Documentum adapter is available out-of-the-box with Federated Search Services. You do not need to install any other application or files. You need to create an adapter backend and configure it as described in [Setting the Documentum Adapter, page 465](#).

Before configuring the Documentum adapter for a particular repository, set its connection configuration to the docbroker as described in the following procedure:

To set the connection to the docbroker:

1. Create a `dfc.properties` file in

`<FS2 installation directory>\lib\jars\config\`

2. Perform one of the following actions:

- Set the following parameters to indicate the host and port of the docbroker:

`dfc.docbroker.host[x]=<host_name> dfc.docbroker.port[x]=<port_number>`

where *x* is the number of the entry. There is one entry per docbroker.

Perform this action when no DFC is installed on the machine or to use a docbroker different from the one of the global DFC.

- Replace the content of the file with the following line to point to an existing `dfc.properties`, for example,

`#include C:/Documentum/Config/dfc.properties`

Perform this action to use the parameters in the `dfc.properties` file of a DFC already installed on the machine.

Updating the Documentum Adapter

By default, the Documentum adapter is updated automatically from the EMC FTP site. To update the adapter manually, download the compressed file available on the EMC Online Support (<https://support.emc.com>).

The `FS2_Adapter_core_DocumentumFC.zip` file includes the following:

- `DocumentumFC.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\core\DocumentumFCbackends\`

- `source-documentum.gif` in `<FS2 installation directory>\www\tomcat\webapps\ao\data\sources\`
- `core_DocumentumFC.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`

Unzip the `FS2_Adapter_core_DocumentumFC.zip` file in the Federated Search Services installation directory.

Setting the Documentum Adapter

This section describes how to create the adapter backend on your FS2 server and how to configure it.

Creating a Backend for the Documentum Adapter

The following procedure describes the steps to add an adapter backend for the Documentum adapter.

To create a backend configuration

1. Log in to FS2 Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend.
5. In the Intranet source list, select **DocumentumFC**.
6. Follow the configuration wizard to set up the backend. The section [Configurable Properties](#), page 465 provides information about the configurable properties of the backend.

Alternatively, you can also use a preexisting configuration (**Select existing information source** option) available in the directory:

```
<FS2 installation directory>\www\wrappers\core\documentumFCBackends\
```

Configurable Properties

The Documentum adapter properties can be configured during the backend creation.

Mandatory Properties

This section describes the mandatory properties for the Documentum adapter.

Table 72. Documentum adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers).
Default value	core/DocumentumFC.jar

Table 73. Documentum adapter properties — baseName

Item	Description
Description	The name of the repository to connect to (for example dm_my_repository)
Default value	None

Table 74. Documentum adapter properties — docType

Item	Description
Description	The name of the Documentum collection on which the search is performed. It must be a subtype of dm_sysobject (for example, dm_document, dm_folder, dm_method).
Default value	dm_document

Optional Properties

This section describes the optional properties for the Documentum adapter.

Table 75. Documentum adapter properties — host

Item	Description
Description	Hostname to include in the URL of the document. Usually, it is the name of the Documentum server.
Default value	None

Table 76. Documentum adapter properties — attributes

Item	Description
Description	<p>Define the list of Documentum attribute names that are sent by Documentum gateway for every search. Whatever the attributes defined in the query, the Documentum server displays this list of attributes for each results.</p> <p>Note: This property is critical regarding the behavior of the Documentum adapter.</p> <p>For example, you can add <code>i_folder_id</code> to associate the folder location to each document.</p>
Default value	<code>object_name, title, subject, keywords, authors, r_creation_date, r_modify_date, r_content_size, a_content_type, owner_name</code>

Table 77. Documentum adapter properties — client.overview

Item	Description
Description	<p>A comma-separated list of attributes which constitutes the overview field in the HTML interface.</p> <p>Not currently exposed in client applications.</p>
Default value	<code>keywords, author</code>

Table 78. Documentum adapter properties — constraint

Item	Description
Description	Extra constraint to add to the “where” clause of the DQL query.
Default value	None

Table 79. Documentum adapter properties — optimized

Item	Description
Description	<p>Optimize DQL for 5.2 repositories and above.</p> <p>Set to false for legacy repositories that return multiple rows for repeating attributes.</p>
Default value	<p>Inherited from adapter bundle</p> <p>true</p>

Table 80. Documentum adapter properties — flushResults

Item	Description
Description	Incrementally return results to client by chunks.
Default value	100

Table 81. Documentum adapter properties — stopLimit

Item	Description
Description	Maximum number of results returned by this adapter, after FS2 filtering. If too many results are filtered, verify the value of the <i>optimizedStopLimit</i> parameter.
Default value	50

Table 82. Documentum adapter properties — optimizedStopLimit

Item	Description
Description	The value of the DQL hint. It corresponds to the maximum number of results returned by the source before FS2 filtering, for this reason, it must be higher than the <i>stopLimit</i> property. For repositories version 6.5 SP2 or above, the property is set to: SQL_DEF_RESULT_SET 60, OPTIMIZE_TOP 60 to avoid duplicate results and performance issues. For repositories version 6.5 SP1 or below, set the property to: SQL_DEF_RESULT_SET 60, RETURN_TOP 60, OPTIMIZE_TOP 60
Default value	SQL_DEF_RESULT_SET 60, OPTIMIZE_TOP 60

Table 83. Documentum adapter properties — orderClause

Item	Description
Description	Add ORDER BY clause at the end of the WHERE clause For example, ORDER BY r_creation_date DESC
Default value	None

Table 84. Documentum adapter properties — preferredRendition

Item	Description
Description	This property applies when the user requests to view a document. The document is returned in the following rendition if available. If this rendition is not available, the property secondaryRendition is used. The rendition corresponds to the name of the format in the repository: msw8 (for Word), pdf, text.
Default value	None

Table 85. Documentum adapter properties — secondaryRendition

Item	Description
Description	<p>This property applies when the user requests to view a document and when the document is not available in the preferred rendition. The document is returned in the following rendition if available. If this rendition is not available, the original document is returned.</p> <p>The rendition corresponds to the name of the format in the repository: msw8 (for Word), pdf, text.</p>
Default value	None

Table 86. Documentum adapter properties — dateFormat

Item	Description
Description	Format of dates in the repository.
Default value	MM/dd/yyyy

Table 87. Documentum adapter properties — viewUrl

Item	Description
Description	<p>URL redirection. Set a Webtop-base URL value to redirect the document request to Webtop, that is, build a fixed URL instead of viewing document from the Content Server (through DFC). With viewURL set, each document URL is:</p> <p><code><viewURL>+<objectId></code> For example, if you set viewUrl=http://<machine name>:<port>/webtop/drl/objectId/, a possible URL is: http://webtophost:8100/webtop/drl/objectId/0900584a80001398.</p>
Default value	None

Table 88. Documentum adapter properties — filter

Item	Description
Description	Set to true to post-filter results before sending them to the client.
Default value	None

Table 89. Documentum adapter properties — useFTI

Item	Description
Description	Set to true if the repository is full-text indexed and if you want to use the full-text attribute. If it is set to true do not forget to add body as a value of the property trusted in order to not remove valid results.
Default value	false

Table 90. Documentum adapter properties — map.full-text

Item	Description
Description	<p>Defines a list of Documentum attribute names to map the FS2 attribute full-text.</p> <p>This property is ignored when useFTI is set to true.</p> <pre>map.full-text=title,ANY keywords,subject</pre> <p>With such a mapping, if the original query is 'full-text, contains, knowledge', the Documentum query is 'title,contains,Knowledge OR ANY keywords,contains,Knowledge OR subject,contains,Knowledge'.</p> <p>Note: This property is only available for the Documentum adapter.</p>
Default value	object_name, title, subject, ANY keywords, ANY authors, owner_name

Table 91. Documentum adapter properties — image

Item	Description
Description	<p>Special logo used to represent results from this backend; if the value of this property is "XX"; store the logo in: \www\tomcat\webapps\ao\data\sources\source-XX.gif</p>
Default value	documentum

Table 92. Documentum adapter properties — loginName

Item	Description
Description	<p>Name of the guest user.</p> <p>If this property and the property loginPassword are set, they are used for every FS2 user to log in to the repository.</p>
Default value	None

Table 93. Documentum adapter properties — loginPassword

Item	Description
Description	<p>Password of the guest user.</p> <p>If this property and the property loginName are set, they are used for every FS2 user to log in to the repository.</p>
Default value	None

Table 94. Documentum adapter properties — supportsLogin

Item	Description
Description	Set to true as the Documentum API allows users to authenticate themselves when performing a search. When the end user does not specify any login, the guest login (loginName and loginPassword) is used.
Default value	true

Table 95. Documentum adapter properties — mapin.<Documentum attributes>

Item	Description
Description	<p>Defines the translation of a Documentum attribute, available in the results, in an FS2 attribute.</p> <p>Syntax:</p> <pre>mapin.<Documentum attribute>=<FS2 attribute></pre> <p>For example,</p> <pre>mapin.r_modify_date=last_modified_date mapin.authors=author</pre> <p>Note: This property is mandatory for this adapter, especially for Documentum repeating attributes. However, do not add ANY before the attribute name.</p>
Default value	None

Table 96. Documentum adapter properties — mapmerge

Item	Description
Description	<p>Defines a list of FS2 attributes that corresponds to a list of Documentum attributes for the queries and the results. This property is linked to the mapmerge.<FS2 attributes> property.</p> <p>Syntax:</p> <pre>mapmerge=<FS2 attributes></pre> <p>Note: This property is optional and it is only available for the Documentum adapter.</p>
Default value	None

Table 97. Documentum adapter properties — mapmerge.<FS2 attributes>

Item	Description
Description	<p>Defines the list of Documentum attributes that correspond to the FS2 attributes defined in the mapmerge property and the type of merge between them, exclusive Or represented by the vertical bar , or And represented by the plus sign +.</p> <p>Syntax:</p> <pre>mapmerge.<FS2 attribute>=<Documentum attribute> <Documentum attribute> mapmerge.<FS2 attribute>=<Documentum attribute>+<Documentum attribute></pre> <p>mapmerge=title,author</p> <pre>mapmerge.title=title object_name</pre> <p>For a query defined in FS2 on the attribute title, such as 'title, CONTAINS, java', the query sent to Documentum is on both attributes title and object_name: 'title, contains, java OR object_name, contains, java'. For the results, if title is defined in Documentum, the title value in FS2 is the same as Documentum title value, otherwise, the title value in FS2 contains the object_name value sent by Documentum.</p> <pre>mapmerge.author=authors+owner_name</pre> <p>A query defined in FS2 on the attribute author is sent on both attributes authors and owner_name in Documentum. For the results, the author value is the list of authors sent by Documentum concatenated with the owner_name value.</p> <p>Note: This property is optional and it is only available for the Documentum adapter.</p>
Default value	None

Table 98. Documentum adapter properties — mapout.<FS2 attributes>

Item	Description
Description	<p>Defines the translation of an FS2 attribute in a Documentum attribute for querying the source.</p> <p>Syntax:</p> <pre>mapout.<FS2 attribute>=<Documentum attribute></pre> <p>For example,</p> <pre>mapout.object_name=object_name mapout.author=ANY authors</pre> <p>Note: This property is mandatory for this adapter, especially for Documentum repeating attributes (that is, attributes that need ANY before the attribute name).</p>
Default value	None

Table 99. Documentum adapter properties — port

Item	Description
Description	Port number to include in the URL of the document.
Default value	80

Table 100. Documentum adapter properties — query

Item	Description
Description	The list of attributes available for querying this source.
Default value	title, author, abstract, keywords, object_name, date, full-text

Table 101. Documentum adapter properties — result

Item	Description
Description	The list of known attributes returned by this source.
Default value	title, author, abstract, keywords, object_name, date, last_modified_date, size, URL, site

Table 102. Documentum adapter properties — trusted

Item	Description
Description	FS2 does not make any filtering on this list of attributes. Often used when the source sends only partial results (that is, only ten first words). This list MUST contain 'body' when 'useFTI' is set to true (that is, the Documentum base is full-text indexed).
Default value	None

Documentum eRoom Adapter

The Documentum eRoom[®] adapter enables an FS2 client to query the eRoom server in order to consult stored items and documents. Users can visualize the documents selected. If you implement the eRoom web service, users can also import the files in the client application, such as Webtop and CenterStage. The import functionality is not available for eRoom objects. When trying to import an eRoom object, an eRoom session is started or, if a session is already started, the eRoom object is opened.

Principles of the eRoom Adapter

The eRoom adapter offers two possible technologies to connect to eRoom server:

- For eRoom version 7.3.495.131¹ and above, we recommend using the adapter based on the eRoom Web Services (a.k.a. XML API).
- For all previous versions of eRoom version 7, the eRoom adapter must use an ASP script stored in the eRoom server.

The eRoom server executes the script in order to obtain the query description and to consult its items. This script then returns the data extracted to the FS2 server and the FS2 server wraps the results and displays them to the end user.

Installing the eRoom Adapter

This section describes the methods to install the eRoom adapter, depending on the eRoom version:

- The adapter installation for eRoom versions *previous to 7.3.495.131* is described in [Installing the FS2 Script for eRoom, page 474](#).
- The adapter installation for eRoom *version 7.3.495.131 and above* is described in [Installing the Web Service-Based Adapter, page 475](#).

Installing the FS2 Script for eRoom

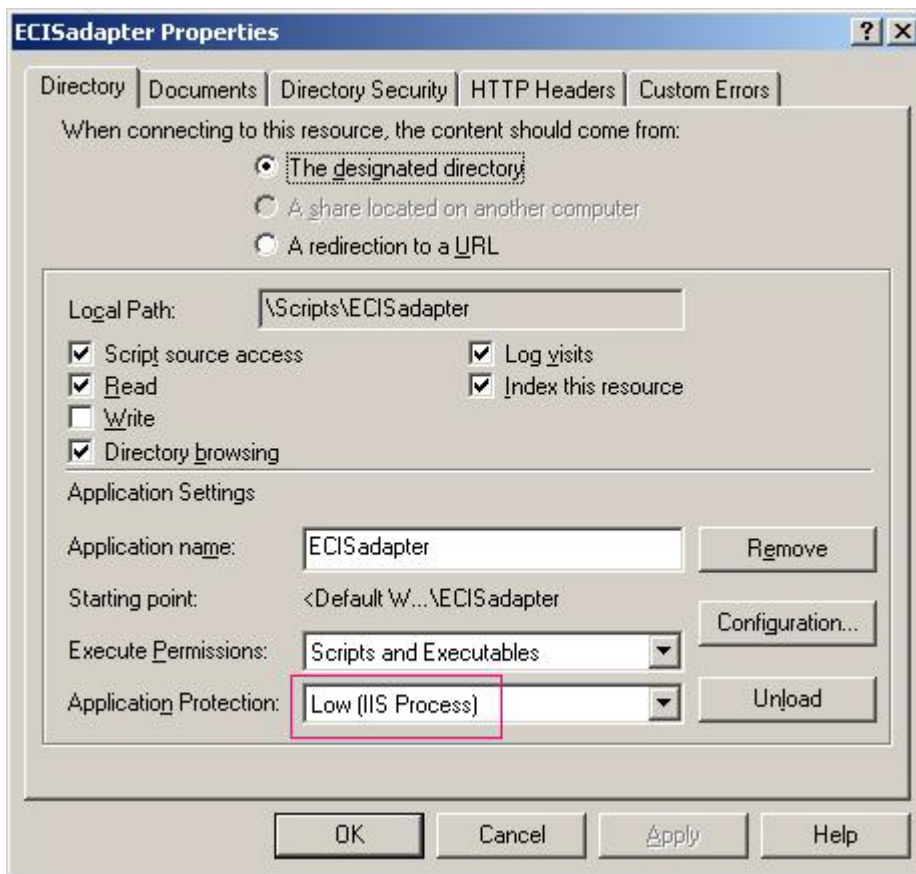
The following procedure describes how to install the FS2 ASP script for the eRoom adapter.

To install eRoom ASP script:

1. Locate the eRoom script: `<FS2 installation directory>/install/adapters/eRoom/ECISearch_eRoom7.asp`
2. Create a specific directory on the eRoom server for the FS2 script for eRoom. For example, `C:\Inetpub\scripts\ECISAdapter`.
3. Copy the script file in the scripts folder.
4. Use Internet Information Services (IIS) Manager to create an alias, such as `ECISAdapter`, for the IIS Server on this directory, with read and script access rights as described in [Figure 18, page 475](#).

1. To find the version number of your eRoom site, go to the home page of your eRoom site and check the **Site settings**. You need eRoom administrator's rights to access these parameters.

Figure 18. IIS server configuration



5. The script contains a form to test queries against eRoom. If the trace option is enabled, the page displays a list of eRooms available in the facility.

Installing the Web Service-Based Adapter

Since the eRoom adapter is available out-of-the-box with Federated Search Services, you do not need to install any other application or files. To install the Web Service (WS)-based adapter, create an adapter backend and configure it as described in [Setting the eRoom Adapter, page 476](#).

The adapter searches the eRoom Server through its Query Service web service that can be reached at `http://<host>/eroomxml?wsdl`.

eRoom web services must be activated on the eRoom Server to allow communication with the adapter. To do so, go to the eRoom **Site Settings** page and select the **allow XML queries and commands from external applications** option.

All the searches are limited to 250 results. If your query retrieves more results, the eRoom server does not return results. Federated Search Services throws a `TOO_MANY_RESULTS` error message.

Updating the eRoom Adapter

The eRoom adapter is automatically installed with Federated Search Services. By default, adapters are updated automatically from EMC FTP site. If you disabled this option, to update manually the adapter, download the compressed file available on EMC Online Support (<https://support.emc.com>).

The FS2_Adapter_core_ERoomWS.zip file includes the following:

- ERoomWS.jar in *<FS2 installation directory>\www\wrappers\core*
- source-eroom.gif in *<FS2 installation directory>\www\tomcat\webapps\ao\data\sources*
- core_ERoomWS.xml in the following locations:
 - *<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template*
 - *<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template*

Note: Unzip the FS2_Adapter_core_ERoomWS.zip file in the FS2 installation directory.

Setting the eRoom Adapter

This section describes how to create the adapter backend on your FS2 server and how to configure it.

Creating an Adapter Backend for eRoom

The following procedure describes the steps to add a new adapter backend for the eRoom adapter.

To create a backend configuration

1. Log in to FS2 Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend.
5. In the Intranet source list, select one of the following:
 - For eRoom version 7.3.495.131 and above, select **ERoomWS** to use the Web Service-based adapter.
 - For previous versions, select **eRoom** to use the ASP script.
6. Follow the configuration wizard to set up the backend. The section [Configurable Properties, page 477](#) provides information about the configurable properties of the backend.

Alternatively, you can also use a preexisting configuration (**Select existing information source** option) available in the directory:

<FS2 installation directory>\www\wrappers\core\eRoombackends

Configurable Properties

The eRoom adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the eRoom adapter.

Table 103. eRoom adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers).
Default value	For ASP-based adapter: core/eRoom.jar For WS-based adapter: core/ERoomWS.jar

Table 104. eRoom adapter properties — host

Item	Description
Description	The name of the server hosting the eRoom.
Default value	None

Table 105. eRoom adapter properties — action (only ASP-based adapter)

Item	Description
Description	Relative URL address for the search interface of the target data source, that is, the ECISearch.asp page in its aliased directory. For example, ECISAdapter/ECISearch.asp
Default value	Scripts/ECISadapter/ECISearch_eRoom7.asp

Table 106. eRoom adapter properties — facilityName

Item	Description
Description	The eRoom facility to be searched. Use the URL name of the facility, not the display name. If specified, set the eRoomName list property. If not specified, the search goes through the entire eRoom site (all facilities and rooms). For example, facilityName=myFacility

Item	Description
Default value	None
Mandatory	Yes (for ASP-based adapter) No (for WS-based adapter)

Table 107. eRoom adapter properties — eRoomName

Item	Description
Description	Comma-separated list of eRooms to be searched in the facility specified by the facilityName property. Use the URL name of the eRoom, not the display name. For example, eRoomName=ClientEngagementV6,LeadGeneration,myRoom
Default value	None
Mandatory	Yes when the facilityName property is defined (for WS-based adapter). No (for ASP-based adapter), even if the facilityName property is defined.

Optional Properties

This section describes the optional properties for the eRoom adapter.

Table 108. eRoom adapter properties — createTracefiles (only WS-based adapter)

Item	Description
Description	Debug option to diagnose issues during the backend creation. When set to true, the adapter creates three files: <ul style="list-style-type: none"> The query sent to the eRoom XML api (ECI_eRoomQuery.xml). The results received (ECI_eRoomResult.xml). The results transformed for extraction by the adapter (ECI_eRoomEciResult.xml). These files can help the support team, if need be.
Default value	False

Table 109. eRoom adapter properties — traceFilesPath (only WS-based adapter)

Item	Description
Description	Folder in which the debug files are created
Default value	C:\temp\

Table 110. eRoom adapter properties — client.overview

Item	Description
Description	A comma-separated list of attributes which constitutes the overview field in the client interface. Not currently exposed in client applications.
Default value	Inherited from adapter bundle: eRoom, author, modified_by

Table 111. eRoom adapter properties — filter

Item	Description
Description	Set to true to post-filter the results before sending them to the client (true/false). The server simply returns all results returned by the source.
Default value	Inherited from adapter bundle: true

Table 112. eRoom adapter properties — image

Item	Description
Description	Special logo used to represent results from this backend.
Default value	eRoom

Table 113. eRoom adapter properties — supportsLogin

Item	Description
Description	Set to true when the backend allows individual users to log in to perform a search on private or confidential data.
Default value	Inherited from adapter bundle: true

Table 114. eRoom adapter properties — loginName

Item	Description
Description	The default login name to use when authenticating with the source. This default login name is used for any end user that did not specify a personal one for this backend. Active only when the property supportsLogin is set to true.
Default value	None

Table 115. eRoom adapter properties — loginPassword

Item	Description
Description	The default login password to use when authenticating with the source. This default login password is used for any end user that did not specify a personal one for this backend. Active only when the property supportsLogin is set to true.
Default value	None

Table 116. eRoom adapter properties — port

Item	Description
Description	The port number of the server.
Default value	Inherited from adapter bundle: 80
Mandatory	Yes for HTTP adapter

Table 117. eRoom adapter properties — protocol

Item	Description
Description	The protocol to use when connecting to the source: http or https. To disable the remote source authentication with https, use the value: https-no-auth. For example, protocol=https or protocol=https-no-auth
Default value	Inherited from adapter bundle: http

Table 118. eRoom adapter properties — proxySet

Item	Description
Description	Set to true to use the default HTTP proxy to access to source.
Default value	Inherited from adapter bundle: false

Table 119. eRoom adapter properties — query

Item	Description
Description	Comma-separated list of primary attributes available for querying the source.
Default value	Inherited from adapter bundle: title, collection, description, is_unread, date, size, full-text (for ASP-based adapter) title, full-text (for WS-based adapter)

Table 120. eRoom adapter properties — result

Item	Description
Description	Comma-separated list of known attributes returned by the source.
Default value	Inherited from adapter bundle: title, record_id, is_unread, version, eRoom, collection, collection_URL, author, creation_date, modified_by, date, site, source, URL, document_type, description (for ASP-based adapter) title, record-id, eRoom, collection, author, creation_date, date, site, source, URL, type, format (for WS-based adapter)

Table 121. eRoom adapter properties — trusted

Item	Description
Description	The list of attributes that are trusted. That is to say that no filtering is done on these attributes. It is useful when a source does not return the full content of an attribute. We can “trust” the selection and filtering of results implemented by the source. Note: It also applies to an attribute that is never returned by the source but still is “trusted”. For example, a source supporting a search on “keywords” but that does not return a “keywords” attribute. A search on “full-text” always matches when some attributes are configured as trusted.
Default value	None

Table 122. eRoom adapter properties — stopLimit

Item	Description
Description	Maximum number of results returned by this adapter, after FS2 filtering. If too many results are filtered, verify the value of the <i>optimizedStopLimit</i> parameter.
Default value	50

Table 123. eRoom adapter properties — optimizedStopLimit

Item	Description
Description	Maximum number of results returned by the source before FS2 filtering. Make sure that this value is higher than the stopLimit. This is only supported for site searches; that is, when the properties <i>facilityName</i> and <i>eRoomName</i> are not defined.
Default value	100

Documentum ApplicationXtender Adapter

The ApplicationXtender® adapter makes the server data sources and applications of ApplicationXtender searchable from any FS2 client. The adapter is compatible with the following versions of ApplicationXtender:

- 5.4
- 6.0
- 6.5
- 6.52
- 7.0

The adapter interacts with ApplicationXtender through the ApplicationXtender web services.

Principles of the ApplicationXtender Adapter

The adapter searches the ApplicationXtender Server through ApplicationXtender Web Services that can be reached at: `http://<host>:<port>/AppXtender/AxServicesInterface.asmx?WSDL`

ApplicationXtender Web Services (AppXtender Web Services) is a set of web services that provides a remote interface for accessing the ApplicationXtender Content Management system. AppXtender Web Services has to be in integrated mode. In integrated mode, AppXtender Web Services and AppXtender Web.NET can share the same session and documents can be displayed using browser control in AppXtender Web.NET according to the current user profile settings. The *EMC ApplicationXtender Web Services Administrator Guide* provides more information about ApplicationXtender settings.

The ApplicationXtender adapter only supports ApplicationXtender system that uses ProIndex for full-text indexing.

Source Software Requirements

The source software has to be ApplicationXtender 5.4, 6.0, 6.5, 6.52, and 7.0.

The required ApplicationXtender components are:

- AppXtender Web Services
- AppXtender ProIndex full-text engine
- AppXtender Rendering Server

Installing the ApplicationXtender Adapter

Since the ApplicationXtender adapter is available out-of-the-box with Federated Search Services, you do not need to install any other application or files. Simply create an adapter backend and configure it as described in [Setting the ApplicationXtender Adapter, page 483](#).

Updating the ApplicationXtender Adapter

The ApplicationXtender adapter is automatically installed with Federated Search Services. By default, adapters are updated automatically from EMC FTP site. If you disabled this option, to update manually the adapter, download the compressed file available on EMC Online Support (<https://support.emc.com>).

The FS2_Adapter_core_AX.zip file includes the following:

- AX.jar in `<FS2 installation directory>\www\wrappers\core\`
- source-ax.gif in `<FS2 installation directory>\www\tomcat\webapps\ao\data\sources\`
- core_AX.xml in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`
- AXStubs.jar, AXWS.jar, and commons-logging.jar in `<FS2 installation directory>\lib\wrapper\`

Unzip the FS2_Adapter_core_AX.zip file in the FS2 installation directory.

Setting the ApplicationXtender Adapter

This section describes how to create the adapter backend on your FS2 server. One backend is dedicated to one ApplicationXtender Application.

Creating an adapter backend for ApplicationXtender

The following procedure describes how to create an adapter backend for an ApplicationXtender adapter.

To create an adapter backend for an ApplicationXtender adapter:

1. Choose a host directory for your backend. This directory must be under `<FS2 installation directory>\www\wrappers\`. You can choose to use the one that exists, or create a new one. For example, the directory can be: `<FS2 installation directory>\www\wrappers\core\AXbackends`.
2. Create the entry for your backend as described in the *EMC Documentum Federated Search Services Administration Guide*.
3. Add/create the configuration files for the new backend. For each adapter, at least two files are required.
For example, if we want to consult the ApplicationXtender application called application1, the name of the backend file could be AXApplication1.conf. Therefore, the two files are AXApplication1.conf and AXApplication1_en.properties.

4. If you are using a locale, add the localized properties file, for example, `AXApplication1_fr.properties` for the French locale.
5. You can set other properties as described in the *EMC Documentum Federated Search Services Administration Guide*.

Configurable Properties

This section presents the common properties that you can use to configure an ApplicationXtender adapter backend.

Mandatory Properties

This section describes the mandatory properties for the ApplicationXtender adapter.

Table 124. ApplicationXtender adapter properties — action

Item	Description
Description	Location of the Web Services on the remote ApplicationXtender server.
Default value	/AppXtender/AxServicesInterface.asmx?WSDL

Table 125. ApplicationXtender adapter properties — dataSourceName

Item	Description
Description	The data source name on which you perform the search. For example, <code>dataSourceName=myDataSource</code> .
Default value	None

Table 126. ApplicationXtender adapter properties — applicationName

Item	Description
Description	The ApplicationXtender application on which you perform the search. For example, <code>applicationName=HR</code>
Default value	None

Table 127. ApplicationXtender adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (<code>www/wrappers</code>) using the <code>'/'</code> character as the path separator. It consists of <code><domain name>/<backend_name>.jar</code> .
Default value	<code>core/AX.jar</code>

Table 128. ApplicationXtender adapter properties — host

Item	Description
Description	The hostname of the ApplicationXtender server (on which the Web Services are running); an IP number or a valid DNS name.
Default value	None

Optional Properties

This section describes the optional properties for the ApplicationXtender adapter.

Table 129. ApplicationXtender adapter properties — client.overview

Item	Description
Description	A comma-separated list of attributes that is displayed in the overview column of the client interface. Not currently exposed in client applications.
Default value	application_name,date,source

Table 130. ApplicationXtender adapter properties — dateFormat

Item	Description
Description	The date format of the source that is used to convert dates from the FS2 date format to the format of the source.
Default value	MM-dd-yyyy

Table 131. ApplicationXtender adapter properties — filter

Item	Description
Description	Set to true to post-filter the results before sending them to the client (true/false). If set to false, the server simply returns all results returned by the source.
Default value	true

Table 132. ApplicationXtender adapter properties — image

Item	Description
Description	Special logo used to represent results from this backend. If the value of this property is "ACME" (for example, image=ACME), store the logo in: <code>\www\tomcat\webapps\ao\data\sources\source-acme.gif</code> . Note: Image name must always be in lowercase.
Default value	ax

Table 133. ApplicationXtender adapter properties — supportsLogin

Item	Description
Description	Set to true as the ApplicationXtender API allows users to authenticate themselves when performing a search. When the end user does not specify any login, the default login is used.
Default value	true

Table 134. ApplicationXtender adapter properties — loginName

Item	Description
Description	<p>The default login name to use when authenticating with the source. This default login name is used for all end users that did not specify a personal one for this backend.</p> <p>Active only when the property supportsLogin is set to true.</p>
Default value	None

Table 135. ApplicationXtender adapter properties — loginPassword

Item	Description
Description	<p>The default login password to use when authenticating with the source. This default login password is used for any end user that did not specify a personal one for this backend.</p> <p>Active only when the property supportsLogin is set to true.</p>
Default value	None

Table 136. ApplicationXtender adapter properties — mapin.<ApplicationXtender attributes>

Item	Description
Description	<p>The list of mapin.<<i>ApplicationXtender attribute</i>> properties defines how an ApplicationXtender application attribute name (that is, one of the row names) is translated in an internal FS2 attribute name.</p> <p>For example, mapin.NAME=title means that, when NAME is received from the remote source, it is added in FS2 as title.</p> <p>If you are not sure of the name, leave this value empty and run a test using FS2 Admin Center. It will display the list of available fields for the mapping.</p>
Default value	None

Table 137. ApplicationXtender adapter properties — mapout.<FS2 attributes>

Item	Description
Description	<p>The list of mapout.<FS2 attributes> properties defines how an FS2 attribute is translated into an attribute of the source (that is, the name of the corresponding row).</p> <p>For example, mapout.title=NAME means that title is sent to the source as NAME.</p> <p>If you are not sure of the name, leave this value empty and run a test using FS2 Admin Center. It will display the list of available fields for the mapping.</p>
Default value	None

Table 138. ApplicationXtender adapter properties — port

Item	Description
Description	The port number of the server
Default value	80

Table 139. ApplicationXtender adapter properties — proxySet

Item	Description
Description	Set to true to use the default HTTP proxy to access to source.
Default value	false

Table 140. ApplicationXtender adapter properties — query

Item	Description
Description	Comma-separated list of primary FS2 attributes available for querying the source. The attributes defined should match the mapin/mapout properties lists.
Default value	title,full-text,date

Table 141. ApplicationXtender adapter properties — result

Item	Description
Description	The list of known FS2 attributes returned by this source.
Default value	title, application_name, source, URL

Table 142. ApplicationXtender adapter properties — stopLimit

Item	Description
Description	Maximum number of results displayed to the end user, after FS2 filtering. If too many results are filtered, verify the value of the <i>optimizedStopLimit</i> parameter.
Default value	50

Table 143. ApplicationXtender adapter properties — optimizedStopLimit

Item	Description
Description	Maximum number of results returned by the source before FS2 filtering. Make sure that this value is higher than the stopLimit.
Default value	75

Table 144. ApplicationXtender adapter properties — titleFrom

Item	Description
Description	Defines how the FS2 required title attribute is created from ApplicationXtender application fields. It can be a single attribute or a list of attributes. It is not required if the title FS2 attribute is set with mapin/mapout properties. Example : titleFrom=first_name,last_name
Default value	None

Table 145. ApplicationXtender adapter properties — trusted

Item	Description
Description	The list of attributes that are trusted. That is to say that no filtering is done on these attributes. It is useful when a source does not return the full content of an attribute. We can still “trust” the selection and filtering of results implemented by the source. Note: It also applies to an attribute that is never returned by the source but still is “trusted”. For example, a source supporting a search on “keywords” but that does not return a “keywords” attribute. A search on “full-text” always matches when some attributes are configured as trusted.
Default value	None

Query Translation

The FS2 translator converts a query into an AppXtender ProIndex query. This section describes how the queries are translated. The ApplicationXtender adapter supports two types of queries: fields and full-text searches. Fields searches and Full-text searches work together.

Full-Text Searches

Expression searches are supported. The operators AND, OR, NOT, and NEAR(distance) can be used to define the search expression.

Wildcards are supported for all operators but NEAR.

Phrases are not supported and should be defined as a search expression.

The following table illustrates some examples of full-text searches.

Table 146. ApplicationXtender adapter — Full-text searches

FS2 search	AX search
full-text:CONTAINS:deposits AND account	deposits* AND account*
full-text:CONTAINS:deposits NEAR account	deposits NEAR(10) account
full-text:CONTAINS: (deposits AND account) AND (withdrawals NEAR credit)	((deposits* and account*) and (withdrawals near(10) credit))

Since phrases are not supported, you can use the NEAR operator to simulate phrases. For example, searching for:

"This agreement will remain" AND payroll
should be turned into:

(this NEAR(1) agreement) AND (will NEAR(1) remain) AND payroll

The following expression is simpler but a little wider, it would therefore return the expected results but with some noise:

this AND agreement AND will AND remain AND payroll




Field Searches

Field searches do not support Boolean and proximity (near) operators. Phrases are not supported as well.

Wildcards are supported for all the fields but the user-defined list fields.

The BEFORE, AFTER, EQUALS, and BETWEEN operators can be used with date and numerical query attributes. The following figure shows a field search in ApplicationXtender.

Figure 19. AppXtender Web Access .NT search

Show	Index Name	Search Value	
<input checked="" type="checkbox"/>	EMPLOYEE ID	<input type="text"/>	
<input checked="" type="checkbox"/>	LAST NAME	<input type="text"/>	
<input checked="" type="checkbox"/>	FIRST NAME	<input type="text"/>	

Searching user-defined list fields is possible, but the search terms must exactly match, which means that the end user knows the exact list of terms in the list. [Figure 20, page 490](#), shows a user-defined list field in ApplicationXtender. [Table 147, page 490](#), illustrates some examples of field searches.

Figure 20. ApplicationXtender user-defined list field

<input checked="" type="checkbox"/>	DOC. TYPE	<input type="text" value="*"/> <div> * {Null} APPLICATION BACKGROUND AUTHORIZATION CHANGE FORM COMPUTER USE POLICY DIRECT DEPOSIT AGREEMENT DRUG SCREEN EMAIL USE POLICY EMPLOYEE PHOTOGRAPH I9 FORM </div>
-------------------------------------	-----------	--

Table 147. ApplicationXtender adapter — Field searches

FS2 search	AX search
field:CONTAINS:crowley	field: List : '*crowley*'
first_name:CONTAINS: susana OR jason OR philipp OR christina OR katherine / job_code:>=:2920	FIRST_NAME field:List: '*susana*', '*jason*', '*philipp*', '*christin*', '*katherin*' JOB_CODE field: Expression: >= 2920 - ID=7
date:BETWEEN: 2002-01-01,2002-12-31	date field: Expression: ['01-01-2002','12-31-2002']
doc_type:CONTAINS:"CHANGE FORM"	Doc_type field: List: CHANGE FORM

Troubleshooting

If you encounter an error such as (401) `Unauthorized error`, check the settings of the WebServices Web Application in Internet Information Services (IIS) Manager. Open the **Properties** window, select **Directory Security > Authentication Methods**. Then select **Enable anonymous access**.

SourceOne Adapter

The purpose of the EMC SourceOne™ adapter is to allow SourceOne users to search their email archives using one of the FS2 clients.

Principles of the SourceOne Adapter

The SourceOne adapter relies on SourceOne Web Services. It requires that full-text indexing is installed and running on SourceOne.

Support of SourceOne 6.5

The SourceOne adapter for SourceOne version 6.5 only supports text searches.

Basic authentication is required, however, unlike SourceOne Search application, the adapter only supports Windows logon type . The Search types are like the ones available in the SourceOne Search application and can be configured for each backend. You also have the possibility to specify the list of folders to define the search scope.

Support of SourceOne 6.6, 6.8, and 7.0

The SourceOne adapter for SourceOne versions 6.6, 6.8, and 7.0 supports date searches and searches on numerical values in addition to text searches.

As for the authentication, two logon types are supported: Windows and Lotus Notes. Like the adapter for SourceOne 6.5, you also have the possibility to specify the list of folders to define the search scope. One limitation for this adapter is that only email can be searched. For example, contacts and calendar events cannot be searched.

Installing the SourceOne Adapter

Since the SourceOne adapter is available out-of-the-box with Federated Search Services, you do not need to install any other application or files. Simply create an adapter backend and configure it as described in [Setting the SourceOne Adapter, page 492](#).

Configuring Internet Information Services (IIS) Manager

To allow the access to search results, make sure that Internet Information Services (IIS) Manager is configured.

In Internet Information Services (IIS) Manager, select the site **ExDocMgmtSvc** and make sure the options **Anonymous Authentication** and **Basic Authentication** are enabled. The other options must be disabled.

Updating the SourceOne Adapter

The SourceOne adapter is automatically installed with Federated Search Services. By default, adapters are updated automatically from EMC FTP site. If you disabled this option, to update manually the adapter, download the compressed file available on EMC Online Support (<https://support.emc.com>).

The `FS2_Adapter_bundleSOURCEONE.zip` file includes the following:

- `SourceOne.jar` (for SourceOne 6.5) and `SourceOne66.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\core\SourceOnebackends\` and `<FS2 installation directory>\www\wrappers\core\SourceOne66backends\`
- Library files in `<FS2 installation directory>\lib\wrapper\` and `<FS2 installation directory>\lib\jars\`
- `source-sourceone.gif` in `<FS2 installation directory>\www\tomcat\webapps\ao\data\sources\`
- `core_SourceOne.xml` (for SourceOne 6.5) and `core_SourceOne66.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`

Unzip the `FS2_Adapter_bundleSOURCEONE.zip` file in the FS2 installation directory.

Note: You can also download and install `FS2_Adapter_core_SourceOne.zip` and `FS2_Adapter_core_SourceOne66.zip` separately.

Setting the SourceOne Adapter

This section describes how to create the adapter backend on your FS2 server and how to configure it.

Creating an adapter backend for SourceOne

The following procedure describes the steps to add a new adapter backend for the SourceOne adapter.

To create a backend configuration

1. Log in to FS2 Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend.
5. In the Intranet source list, select **SourceOne for S1 6.5** or **SourceOne for S1 6.6** depending on the version of SourceOne you use.

6. Follow the configuration wizard to set up the backend. The section [Configurable Properties](#), page 493 provides information about the configurable properties of the backend.

Alternatively, you can also use a preexisting configuration (**Select existing information source** option) available in the directories:

```
<FS2 installation directory>\www\wrappers\core\SourceOneBackends\  
<FS2 installation directory>\www\wrappers\core\SourceOne66Backends\
```

Configurable Properties

The SourceOne adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the SourceOne adapter.

Table 148. SourceOne adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers).
Default value	core/SourceOne.jar (for SourceOne 6.5) or core/SourceOne66.jar

Table 149. SourceOne adapter properties — host

Item	Description
Description	The hostname of the SourceOne server. It can be either an IP address or a valid DNS name. It is the Web Services server. The host that is defined is used to follow the web services link at : <code>http://<host>/SearchWS/ExSearchWebService.asmx</code> .
Default value	None

Table 150. SourceOne adapter properties — domainName

Item	Description
Description	Windows domain name
Default value	None

Table 151. SourceOne adapter properties — searchType

Item	Description
Description	<p>Type of search that users are authorized to run.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • Owner, which corresponds to “My Items” in SourceOne Search types. • Administrator, which is the same search type as in SourceOne. • ReadAll, which corresponds to “All Items” in SourceOne Search types. • Contributor, which corresponds to “My Contributed Items” in SourceOne Search types. <p>Please refer to <i>SourceOne Search User Guide</i> for the description of SourceOne Search types.</p>
Default value	Owner

Table 152. SourceOne adapter properties — sourceList

Item	Description
Description	<p>Comma-separated list of SourceOne folders to search. The folders listed are the folders for which the users have been granted appropriate permissions (refer to the searchType property).</p> <p>For example: Personal Folder, EmailXtender, 3 Year Folder, 1 Year Folder</p>
Default value	None

Table 153. SourceOne adapter properties — tempFolder

Item	Description
Description	This property is only for the SourceOne66 adapter backends. Temporary folder required to store extracted messages.
Default value	None

Optional Properties

This section describes the optional properties for the SourceOne adapter.

Table 154. SourceOne adapter properties — loginName

Item	Description
Description	Name of the guest user. If this property and the property loginPassword are set, they are used for every FS2 user to log in to the source.
Default value	None

Table 155. SourceOne adapter properties — loginPassword

Item	Description
Description	Password of the guest user. If this property and the property loginName are set, they are used for every FS2 user to log in to the source.
Default value	None

Table 156. SourceOne adapter properties — client.overview

Item	Description
Description	A comma-separated list of attributes which constitutes the overview field in the HTML interface. Not currently exposed in client applications.
Default value	recipient,author,size

Table 157. SourceOne adapter properties — image

Item	Description
Description	Special logo used to represent results from this backend; if the value of this property is "XX"; store the logo in: \www\tomcat\webapps\ao\data\sources\source-XX.gif
Default value	sourceone

Table 158. SourceOne adapter properties — dateFormat

Item	Description
Description	Format of dates specified in the regional settings on the SourceOne Search Web Server.
Default value	yyyy-MM-dd

Table 159. SourceOne adapter properties — filter

Item	Description
Description	Set to true to post-filter results before sending them to the client. If set to false, all results are returned.
Default value	true

Table 160. SourceOne adapter properties — query

Item	Description
Description	The list of attributes available for querying this source.
Default value	full-text,title,body,recipient,author,attachment,attachmentname, size,date,hasattachments

Table 161. SourceOne adapter properties — result

Item	Description
Description	The list of known attributes returned by this source.
Default value	title,record-id,size,recipient,author,cc,date,priority,sensitivity,encrypted, folder,type,hasattachments

Table 162. SourceOne adapter properties — trusted

Item	Description
Description	FS2 does not make any filtering on this list of attributes. This is useful when a source does not return the full content of an attribute. We can "trust" the selection and filtering of results implemented by the source. It also applies to an attribute that is never returned by the source but still is "trusted". For example, a source supporting a search on "body" but that does not return a "body" attribute. A search on "full-text" always matches when some attributes are configured as trusted.
Default value	body,attachment,attachmentname

Table 163. SourceOne adapter properties — msgRendition

Item	Description
Description	<p>This property is only for SourceOne 6.5 adapter backends. For SourceOne 6.6 adapter backends, only the native format is supported.</p> <p>This property applies when the user requests to view a document. The document is returned in the selected rendition type:</p> <ul style="list-style-type: none"> html, returns the email as an html document. nativemsg, returns the email in its native format (.msg for Exchange).
Default value	html

Table 164. SourceOne adapter properties — pageSize

Item	Description
Description	Number of results returned to client in one batch. Modify this property to optimize the response time. A low pageSize value will result in more WebService calls.
Default value	25

Table 165. SourceOne adapter properties — stopLimit

Item	Description
Description	Maximum number of results to fetch per query. Make sure that the value set is not higher than the SourceOne "Maximum Results" properties (Refer to the EMC SourceOne Console, Application Configuration, Web Search).
Default value	50

Table 166. SourceOne adapter properties — protocol

Item	Description
Description	The protocol to use when connecting to the source: http or https.
Default value	http

Table 167. SourceOne adapter properties — logonType

Item	Description
Description	This property is only for SourceOne 6.6 adapter backends. Type of logon account. Possible values are "windows" for Exchange/Outlook users, "notes" for Domino/Notes users.
Default value	windows

Query Translation

The Federated Search Services translator is translating an FS2 query into a SourceOne query. The following operators are supported.

Constraints Operators

Table 168. SourceOne constraints operators — AND

Item	Description
Description	Use AND when all listed terms must be included.

Item	Description
FS2 operator	AND
SourceOne operator	AND

Table 169. SourceOne constraints operators — OR

Item	Description
Description	Use OR to include either of the listed terms.
FS2 operator	OR
SourceOne operator	OR

Table 170. SourceOne constraints operators — ANDNOT

Item	Description
Description	Use ANDNOT to exclude a term. It is not supported for address fields (sender, recipient, owner)
FS2 operator	ANDNOT
SourceOne operator	-

Table 171. SourceOne constraints operators — Phrase

Item	Description
Description	Use a phrase to specify an exact string. A phrase is enclosed within double quotes.
FS2 operator	" ... "
SourceOne operator	" ... "

Table 172. SourceOne constraints operators — Wildcard

Item	Description
Description	Use a wildcard to replace one or more characters It is not supported for attachment names or address fields.
FS2 operator	*
SourceOne operator	*

Supported Operators

The SourceOne 6.5 adapter backends support the following operators:

- The CONTAINS and EQUALS operators can be used for text expressions.
- The EQUALS operator can also be used for numerical values.
- The date operators AFTER, BEFORE, BETWEEN and the numerical operators EQUAL_GREATER, EQUAL_LOWER, GREATER, LOWER, and BETWEEN are not supported in this version.

SourceOne 6.6 adapter backends support all the operators previously listed, including the date operators and the numerical operators.

Known Limitation

When documents are opened or saved from the result list, they are temporarily cached on the local file system of FS2 server machine. While these cache files are cleaned regularly, a massive import could take up a proportional part of the disk space. For this reason, we strongly advised against importing many files using the SourceOne adapter.

Troubleshooting

This section describes known issues and provides workaround, when available.

Messages Cannot Be Imported

With Documentum repositories version 6.6 or lower, the mime-type for msg objects is missing from the dm_format table. FS2 unifies results based on the mime-type. It is not possible to open msg results because they are not assigned the correct Documentum type.

To fix this issue, in Documentum Administrator, run the following DQL query:

```
UPDATE dm_format OBJECTS SET mime_type='application/vnd.ms-outlook' WHERE "name"='msg'
```

Empty Messages Are Imported

An occasional import issue can occur with the .NET Framework resulting in an empty message being imported.

To fix this issue, perform the procedure described on MSDN website: <http://msdn.microsoft.com/en-us/library/ms752252%28v=vs.90%29.aspx> to modify the Internet Information Services (IIS) configuration.

Login Name and/or Password Cannot Work When Containing Spaces

The SourceOne adapter does not support login name and password that contain spaces. To resolve this issue, remove the spaces in the login name and password.

JDBC/ODBC Adapter

Java supports a protocol named JDBC (Java Database Connectivity) to communicate and to query a relational database management system (RDBMS). Using JDBC, it is possible to connect Federated Search Services with any RDBMS on the market. More information about JDBC can be found at: <http://java.sun.com/products/jdbc/>.

For RDBMS that do not currently provide a native JDBC driver, it is possible to use an ODBC driver instead by using the Java built-in JDBC/ODBC bridge. ODBC (Open Database Connectivity) is a widely accepted standard to communicate with RDBMS. Most RDBMS provide an ODBC driver.

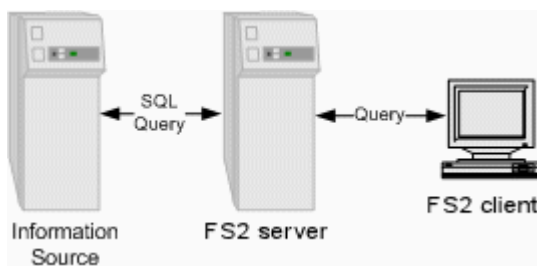
When an RDBMS provides both JDBC and ODBC drivers, it is best to use the native JDBC driver, which provides better performance within Federated Search Services and requires less manual administration work.

Principles of the JDBC/ODBC Adapter

This section describes the capabilities of the JDBC/ODBC adapter.

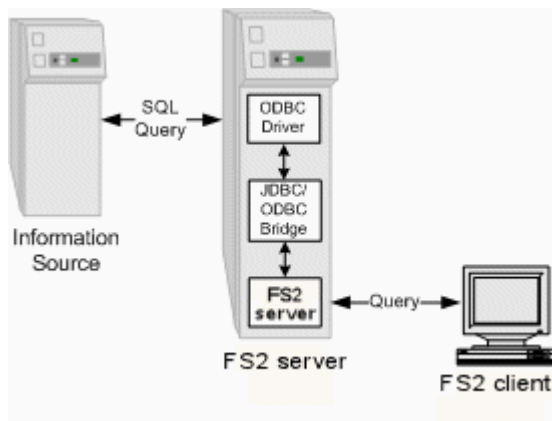
The following figure presents the communication between the participants of a request on a relational database contacted with native JDBC driver. When the user performs a query, the FS2 server uses the JDBC driver to handle the communication with the database possibly located on another computer.

Figure 21. An FS2 query on JDBC



The following figure presents the communication between the participants of a request on a relational database contacted with an ODBC driver through the Java built-in JDBC/ODBC bridge. When the user performs a query, the FS2 server uses the bridge to access the ODBC driver installed on the server machine. The ODBC driver then handles the communication with the database possibly located on another computer.

Figure 22. An FS2 query on ODBC



Installing and Configuring the Database Drivers

There are two ways to add a new adapter talking to an RDBMS: using the ODBC driver (through JDBC/ODBC Bridge) or using a specific JDBC driver for your source. If the database vendor provides a native JDBC driver for Java, we recommend you to use it since it leads to better performance and to a shorter response time for your end users.

Using the JDBC/ODBC Bridge

If you decide to use the ODBC driver, declare your ODBC source on your system.

If you are using Windows NT, the following procedure illustrates this process with a Microsoft Excel database. Otherwise, the ODBC management section of your operating system online help provides more information on how to declare your ODBC source.

To declare your ODBC source for the computer hosting FS2:

1. Launch the Windows NT Control Panel, select the **ODBC data sources** configuration panel, and then select the **System DSN** (Data Source Name).
2. Add a new source by clicking the **Add** button and selecting the driver corresponding to your database, for example Microsoft Access Driver.
3. Configure your ODBC driver with the information specific to your database.

Using a Direct JDBC Driver

Get the JDBC driver for your source. For many information sources, you can download it for free from the Internet, or you can ask your database provider for a JDBC driver for your system. It is a Java library that you can save under `<FS2 installation directory>/lib/jars<MyJDBCDriver>`.

Note: For an Oracle adapter, the previous steps are not required since Federated Search Services ships with the Oracle JDBC native drivers. The `lib\oracle\jdbc\README.txt` provides more information on Oracle driver.

Updating the JDBC/ODBC Adapter

The JDBC/ODBC adapter is automatically installed with Federated Search Services. By default, adapters are updated automatically from EMC FTP site. If you disabled this option, to update manually the adapter, download the compressed file available on EMC Online Support (<https://support.emc.com>).

The `FS2_Adapter_core_Jdbc.zip` file includes the following:

- `Jdbc.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Image files in `<FS2 installation directory>\www\tomcat\webapps\ao\data\sources\`
- `core_Jdbc.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`

Unzip the `FS2_Adapter_core_Jdbc.zip` file in the FS2 installation directory.

Setting the JDBC/ODBC Adapter

This section describes how to create the adapter backend on your FS2 server and how to configure it.

Creating an Adapter Backend for JDBC/ODBC

The following procedure describes the steps to add a new adapter backend for the JDBC/ODBC adapter.

To create a backend configuration

1. Log in to FS2 Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend.
5. In the Intranet source list, select **Jdbc**.
6. Follow the configuration wizard to set up the backend. The section [Configurable Properties, page 503](#) provides information about the configurable properties of the backend.

Configurable Properties

The JDBC/ODBC adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the JDBC/ODBC adapter.

Table 173. JDBC/ODBC adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers).
Default value	core/Oracle.jar (for an Oracle database) core/Jdbc.jar (for all other databases)

Note: The Oracle bundle can be downloaded from EMC Support site (<http://support.EMC.com>).

Table 174. JDBC/ODBC adapter properties — jdbcDriver

Item	Description
Description	<p>Do NOT set this property if you are using the JDBC/ODBC bridge or if you are configuring an Oracle adapter, that is, you must remove the default value.</p> <p>For adapters using a native JDBC driver, this property holds the name of the entry class of the JDBC native driver, for example, com.microsoft.jdbc.sqlserver.SQLServerDriver. This class must be accessible with your CLASSPATH. The documentation of your native JDBC driver provides more details.</p> <p>For adapters using the JDBC/ODBC bridge, this property should be commented out to let the default value active.</p>
Default value	Inherited from adapter bundle: sun.jdbc.odbc.JdbcOdbcDriver

Table 175. JDBC/ODBC adapter properties — jdbcUrl

Item	Description
Description	<p>The source URL, jdbcUrl=jdbc:odbc:<database name in ODBC drivers>, for example, jdbcUrl=jdbc:odbc:db1 (the jdbc URL of the Lotus Notes server)</p> <p>To connect to an Oracle database:</p> <pre>jdbcUrl = jdbc:oracle:thin:@<database_hostname>:<database_port>: <database_name></pre> <p>To connect to a SQLServer database:</p> <pre>jdbcUrl=jdbc:microsoft:sqlserver:// <database_hostname>:<database_port></pre>
Default value	None

Table 176. JDBC/ODBC adapter properties — mapin.<database attributes>

Item	Description
Description	<p>The list of mapin.<database attributes> properties explicitly describes how a database attribute (that is, the row names) is translated in an internal FS2 attribute name.</p> <p>For example, mapin.library1.title=title means that "library1.title" which is received from the remote source is added in FS2 as "title".</p>
Default value	None

Table 177. JDBC/ODBC adapter properties — mapout.<FS2 attributes>

Item	Description
Description	<p>The list of mapout.<FS2 attributes> properties defines how the FS2 attribute is translated into an attribute of the source (that is, the name of the corresponding row).</p> <p>This property is mandatory. The FS2 JDBC adapter uses the association between internal attributes and the attributes of the source when translating the query. Attributes without such association are not considered during the translation phase.</p> <p>For example, mapout.date=table1.publication_date means that date is sent to the source as table1.publication_date.</p>
Default value	None

Note: Always use *mapin.* and *mapout.* notation, even if FS2 attribute has the same name as the field of your database. Otherwise this field is not fetched.

Table 178. JDBC/ODBC adapter properties — fromClause

Item	Description
Description	<p>The FROM clause of the SQL request sent to the source. In your Lotus Notes, if you open the database, the fromClause represents the Forms selected in the design options.</p> <p>In usual RDBMS, this attribute contains the name of the table to search.</p> <p>For example, fromClause=Employees</p>
Default value	None

Table 179. JDBC/ODBC adapter properties — selectClause

Item	Description
Description	<p>The SELECT clause of the SQL request sent to the source. This corresponds to the name of the rows to insert in results.</p> <p>For example, selectClause= publication_date,title,publisher,author,borrowing_status,isbn_issn</p>
Default value	None

Optional Properties

This section describes the optional properties for the JDBC/ODBC adapter.

Table 180. JDBC/ODBC adapter properties — client.overview

Item	Description
Description	<p>A comma-separated list of attributes which constitutes the overview field in the HTML interface.</p> <p>Not currently exposed in client applications.</p>
Default value	<p>Inherited from adapter bundle:</p> <p>abstract,body</p>

Table 181. JDBC/ODBC adapter properties — dateOutputFormat

Item	Description
Description	<p>The format of the date on the source used to convert dates from the FS2 date format to the format of the jdbc source, for example, MM/dd/yyyy. This property is mandatory if your query format could contain FS2 attribute: "date"</p>
Default value	None

Table 182. JDBC/ODBC adapter properties — endClause

Item	Description
Description	The END clause of the SQL request sent to the source. This attribute is optional and can be commented out. For example, endClause=)
Default value	None

Table 183. JDBC/ODBC adapter properties — filter

Item	Description
Description	Set to true to post-filter results before sending them to the client.
Default value	Inherited from adapter bundle: true

Table 184. JDBC/ODBC adapter properties — useFTI

Item	Description
Description	Set to true if the database is full-text indexed. Then, the full-text attribute is mapped according to the value of the property mapout.full-text. You can also add trusted=body. If useFTI is set to false, full-text is mapped to a set of columns according to the property map.full-text.
Default value	Inherited from adapter bundle: true

Table 185. JDBC/ODBC adapter properties — map.full-text

Item	Description
Description	<p>Defines the following list to map full-text (that is, a search on any attributes) to several attributes when the database does not support full-text. The syntax is: map.full-text=<Database attribute>,<Database attribute>.</p> <p>When useFTI is set to false, no index is available. Therefore, a constraint defined on full-text is translated in the attributes connected by 'OR'.</p> <p>For example, if <i>map.full-text = object_name, title, subject, ANY keywords</i>, then <i>full-text, CONTAINS, java</i> in FS2 means one of the following:</p> <ul style="list-style-type: none"> • object_name,CONTAINS,JAVA • title,CONTAINS,JAVA • subject,CONTAINS,JAVA • ANY keywords,CONTAINS,JAVA
Default value	None

Table 186. JDBC/ODBC adapter properties — ignoreCase

Item	Description
Description	Set to true to activate the case sensitivity of the request.
Default value	Inherited from adapter bundle: true

Table 187. JDBC/ODBC adapter properties — image

Item	Description
Description	Special logo used to represent results from this backend. If the value of this property is "XX"; the logo should be stored in: \www\tomcat\webapps\ao\data\sources\source-XX.gif
Default value	www/tomcat/webapps/ao/data/sources/source-<backend_name>.gif

Table 188. JDBC/ODBC adapter properties — keyAttribute

Item	Description
Description	Set this property for Jdbc adapters, which have to treat the accumulation of several rows into one hit.
Default value	None

Table 189. JDBC/ODBC adapter properties — likeMode

Item	Description
Description	A string describing how we translate the CONTAINS operator of the FS2 query. The CONTAINS operator is translated in a like operator that is often LIKE or CLIKE.
Default value	Inherited from adapter bundle: LIKE

Table 190. JDBC/ODBC adapter properties — supportsLogin

Item	Description
Description	Set to true as the backend allows users to authenticate themselves when performing a search. When the end user does not specify any login, the guest login (loginName and loginPassword) is used.
Default value	Inherited from adapter bundle: false

Table 191. JDBC/ODBC adapter properties — loginName

Item	Description
Description	Name of the guest user. This default login name is used for any end user that did not specify a personal one for this backend. Active only when the property supportsLogin is set to true.
Default value	None

Table 192. JDBC/ODBC adapter properties — loginPassword

Item	Description
Description	Password of the guest user. This default login password is used for any end user that did not specify a personal one for this backend. Active only when the property supportsLogin is set to true.
Default value	None

Table 193. JDBC/ODBC adapter properties — proxySet

Item	Description
Description	Set to true to use the default HTTP proxy to access to source.
Default value	Inherited from adapter bundle: false

Table 194. JDBC/ODBC adapter properties — query

Item	Description
Description	The list of primary FS2 attributes available for querying this source.
Default value	Inherited from adapter bundle: title,author,date,publisher,borrowing_status,isbn_issn,full-text

Table 195. JDBC/ODBC adapter properties — query.enclosingDate

Item	Description
Description	The characters that enclosed the date attribute in the SQL query send to the source. It is often ' (mySQL) or # (ODBC) For example: JDBC: SELECT * FROM PeerReviews WHERE (PeerReviews.Date)>#12/12/1999#) mySQL: SELECT * FROM Library1 WHERE publication_date<'01-Oct-1994'
Default value	# (because the default driver is sun.jdbc.odbc.JdbcOdbcDriver)

Table 196. JDBC/ODBC adapter properties — result

Item	Description
Description	The list of known FS2 attributes returned by this source.
Default value	Inherited from adapter bundle: title,author,type,date,publisher,borrowing_status,reference,isbn_issn

Table 197. JDBC/ODBC adapter properties — result.enclosingChar

Item	Description
Description	The characters that enclosed the results given by the source. It is often ' or nothing. For example, result.enclosingChar= '
Default value	None

Table 198. JDBC/ODBC adapter properties — useToDate

Item	Description
Description	Indicates whether the SQL to_date function is available. Set to yes if the SQL to_date function is supported (default). Set to no, the date is sent as-is.
Default value	Inherited from adapter bundle: No

Table 199. JDBC/ODBC adapter properties — whereClause

Item	Description
Description	The beginning of the “where” clause of the SQL request sent to the source. This attribute is optional and can be commented out. For example, dm_sysobject_s.r_object_id_i=dm_sysobject_r.r_object_id_i
Default value	None

Table 200. JDBC/ODBC adapter properties — trusted

Item	Description
Description	The list of attributes that are trusted. That is to say that no filtering is done on those attributes. This is useful when a source does not return the full content of an attribute.
Default value	None

OpenSearch Adapter

The purpose of the OpenSearch adapter is to allow end users to run queries against an OpenSearch source using one of the FS2 clients.

Principles of the OpenSearch Adapter

Opensearch is a description format for search engine plug-ins. It allows publishing of standardized search results from various sources. You need to know the URL to access the OpenSearch search engine. The URL must comply the OpenSearch description syntax. For more information about

the URL syntax, refer to the OpenSearch specifications: http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_3#OpenSearch_URL_template_syntax.

Limitation: this adapter is not suitable for handling large result sets. Indeed, it performs only one request to get all the results and the results are returned in one batch to the client.

Compatibility

The OpenSearch adapter is compatible with OpenSearch 1.0. It also supports OpenSearch version 1.1 (Draft 3).

Installing the OpenSearch Adapter

Since the OpenSearch adapter is available out-of-the-box with Federated Search Services, you do not need to install any other application or files. Simply create an adapter backend and configure it as described in [Setting the OpenSearch Adapter, page 510](#).

Updating the OpenSearch Adapter

The OpenSearch adapter is automatically installed with Federated Search Services. By default, adapters are updated automatically from EMC FTP site. If you disabled this option, to update manually the adapter, download the compressed file available on EMC Online Support (<https://support.emc.com>).

The `FS2_Adapter_indexer_OpenSearch.zip` file includes the following:

- `OpenSearch.jar` in `<FS2 installation directory>\www\wrappers\indexer\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\indexer\OpenSearchbackends\`
- `bing_p.ico`, `source-blogsearch.gif`, and `source-youtube.gif` in `<FS2 installation directory>\www\tomcat\webapps\ao\data\sources\`
- `core_OpenSearch.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`

Unzip the `FS2_Adapter_indexer_OpenSearch.zip` file in the FS2 installation directory.

Setting the OpenSearch Adapter

This section describes how to create the adapter backend on your FS2 server and how to configure it.

Creating an Adapter Backend for OpenSearch

The following procedure describes the steps to add a new adapter backend for the OpenSearch adapter.

To create a backend configuration

1. Log in to FS2 Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend.
5. In the Intranet source list, select **OpenSearch**.
6. Follow the configuration wizard to set up the backend. The section [Configurable Properties](#), page 511 provides information about the configurable properties of the backend.

Alternatively, you can also use a preexisting configuration (**Select existing information source** option) available in the directory:

```
<FS2 installation directory>\www\wrappers\indexer\OpenSearchbackends\
```

Configurable Properties

The OpenSearch adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the OpenSearch adapter.

Table 201. OpenSearch adapter properties — bundle

Item	Description
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (www/wrappers).
Default value	indexer/OpenSearch.jar

Table 202. OpenSearch adapter properties — url

Item	Description
Description	The URL of the OpenSearch search engine, for example, http://api.search.yahoo.com/WebSearchService/rss/webSearch.xml?appid=yahoosearchwebrss&query={searchTerms}&results={count} Make sure the URL complies to OpenSearch specifications. For more information about the URL syntax, refer to the OpenSearch specifications: http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_3#OpenSearch_URL_template_syntax
Default value	None

Optional properties

This section describes the optional properties for the OpenSearch adapter.

Table 203. OpenSearch adapter properties — image

Item	Description
Description	Special logo used to represent results from this backend; if the value of this property is “XX”; store the logo in: <code>www/tomcat/webapps/ao/data/sources/source-XX.gif</code>
Default value	opensearch

Table 204. OpenSearch adapter properties — home

Item	Description
Description	The home page of this information source.
Default value	None

Table 205. OpenSearch adapter properties — stopLimit

Item	Description
Description	Maximum number of results returned by this adapter, after FS2 filtering.
Default value	50

Troubleshooting

Some websites might use OpenSearch partially. For example, Bing.com uses OpenSearch result structure for paging mechanism. If a website is not a standard OpenSearch protocol, you need to customize the OpenSearch adapter. A sample adapter is located in `\fs2\src\custom\training\SampleOpenSearchAdapter\`. You can customize the sample

adapter and build the adapter to handle XML result set. The sample adapter provides detailed information in the Javadoc between the code.

Connecting to a Source Using HTTPS

HTTPS is a secured version of the HTTP protocol. This protocol requires authentication of the remote source before connection is done; it ensures that no other party can intercept or modify the data between Federated Search Services and the remote source.

To use HTTPS, the remote source must be HTTPS-compliant. Therefore, it must propose web pages for which the URL starts with “https://” instead of the standard “http://”. In this case, use the following steps to configure the adapter.

For this section, the following example is used: `https://www.safeweb.com/products.html`

Configuring the Adapter with HTTPS

Open the adapter configuration file (`Safeweb.conf`) and set the properties:

- `host=www.safeweb.com`
- `port=443` (Default port for HTTPS protocol is 443)
- `proxySet=true`
Note: When you are going through a proxy to access the remote source, make sure that the proxy is configured to accept SSL connection.
- `action=/products.html`
- `method=get`
- `protocol=https`

Note: This property declares that the adapter uses HTTPS to connect to the remote source.

Note: Various protocols can be used for different levels by setting the property `protocol<X>` where X is the value of the level. For example:

- `protocol=http`
- `protocol1=https`

These properties declare that globally the protocol to use is HTTP, except for the level 1 page for which the HTTPS protocol is used.

You should at least configure the host, action, and protocol properties.

Wrapping the Remote Source

After the adapter is configured, use `aOWrapperTester` to check if it can connect to the remote source.

For example:

```
<FS2 installation directory>\bin\aoWrapperTester -n training/Safeweb  
-a full-text -q test
```

If the adapter fails and the following error message is printed in the trace: “javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated” then the remote source owns a certificate that is not trusted by Federated Search Services. In this case read the next section to add the certificate to the FS2 certificate store.

Otherwise, if the adapter succeeds or fails with another error message, then the HTTPS connection is correct. When you are using a proxy the message in the trace could be a proxy message.

Note: A special value can be set for the protocol or protocolX properties: protocol=https-no-auth.

This value declares that the FS2 server must NOT ask the remote source to authenticate itself. In this case, the HTTPS security is no more ensured, but on the other hand, there is no need to trust the remote source certificate. Use this special value with special care.

Checking the Remote Source Certificate

To be accessed with HTTPS, the remote source must own a certificate that authenticates itself. The FS2 server must also know and trust this certificate for the connection to be established.

By default, the FS2 server consults a certificate store file that describes the trusted certificates: `<FS2 installation directory>\www\docs\conf\trustedCerts.cer`.

The default certificate store file can be copied in another file to use the new file as a custom certificate store. Several properties can be set in the `<FS2 installation directory>\www\docs\conf\server.conf` file to select other certificate store files, see the property `xtrim.ssl.trustStore` in the *EMC Documentum Federated Search Services Administration Guide*.

The following section describes how to add a new certificate in the FS2 certificate store file:

Saving the Remote Source Certificate

To save the remote source certificate, perform the following steps:

1. Open Microsoft Internet Explorer and enter the URL of the remote source. The URL must start with “https” (for example, `https://www.safeweb.com`) A security alert message box can appear to inform you that you are about to view a secured web page. Accept the message.

If the web page of the remote source is correctly displayed, then the remote certificate is already known by Internet Explorer.

2. Right-click the page, select **Properties**, and then **Certificates**.

Note: The Certification Path information screen appears. Close the window and skip to step 3. Otherwise, a message is displayed warning you that the certificate of the remote source is not trusted. You can click **View certificate** to view it.

3. Click **Install Certificate**.

An installation wizard is displayed, choose the appropriate option to let it automatically select the certificate store and finish.

4. In Internet Explorer, select **Tools > Internet Options**.

5. Select the **Content** tab, and then **Certificate**.
6. Select the appropriate tab corresponding to the store noted in the previous step (Other People, in the example).
7. Select the certificate in the list.
8. Click **Export** to save the certificate in a separate file.

A certificate export wizard appears.

9. For the format of the certificate file, choose DER encoded binary X.509 or Base-64 encoded X.509.
10. Browse the directories, and create a certificate file. For this example, call it safeweb.cer.
11. Save the certificate in *<FS2 installation directory>/www/docs/conf* directory.
12. Complete the steps of the wizard.

The file is created.

13. Close the windows.

The remote source certificate has been saved.

14. Add this certificate to the FS2 trusted certificates store as described in the following section.

Adding the Certificate to FS2 Keystore File

The default FS2 trusted certificates keystore file is *<FS2 installation directory>/www/docs/conf/trustedCerts.cer*. This store file can be managed with the Java utility `keytool` located in `JAVA_HOME/bin/keytool.exe`. The javadoc of the JDK installed on your machine provides more information on the Java utility `keytool`.

To add the certificate to the FS2 keystore file, perform the following steps:

1. If necessary, move the remote source certificate file (created in the previous section) in the directory *<FS2 installation directory>/www/docs/conf*.
2. Open a DOS console, go to this directory, and type the following command:

```
JAVA_HOME/bin/keytool -import -alias <my alias> -file <my
certificate> -keystore <my keystore file>
```

where:

<FS2 installation directory> is the directory in which Federated Search Services is installed.

<my alias> is an alias chosen for the certificate. Any name can be chosen. Type a name that can help you identify the associated certificate.

<my certificate> is the name of the certificate file (created in the previous section).

<my keystore file> is the name of file that contains all the certificates for the trusted remote sources (for example, `trustedCerts.cer`, which is the default trusted certificates keystore file for Federated Search Services). It can be replaced by any other certificate keystore file.

For example,

```
JAVA_HOME/bin/keytool -import -alias safeweb
-file safeweb.cer -keystore trustedCerts.cer
```

In this example, the password of the trustedCerts store is *changeit*.

The keytool utility prompts to confirm the addition of the new certificate.

3. Type **Yes**.

The new certificate to trust is added.

4. Verify that the adapter can connect to the remote source.

Note: The keytool utility provides other commands to manage certificates stores. for example, you can list the certificates of the store. In the directory *<FS2 installation directory>/www/docs/conf*, type the command `JAVA_HOME/bin/keytool -list -keystore trustedCerts.cer` with the password **changeit**.

xPlore adapter installation

Introduction

The EMC Documentum Federated Search Services xPlore adapter makes the xPlore Search Service searchable from any Federated Search Services client such as Webtop. This section provides instructions for installing the xPlore adapter. The *EMC Documentum Platform and Platform Extensions Installation Guide* provides more information about adapters, backend configuration, and common Federated Search Services attributes.

The most up-to-date information about Federated Search Services is available in the *EMC Documentum Platform and Platform Extensions Release Notes*.

This section is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing xPlore Adapter

Installation Requirements

The xPlore adapter supports xPlore 1.6 and earlier versions.

The xPlore adapter is compatible with Federated Search Services 7.3.

Configuring xPlore Server

Complete the following steps to configure the adapter for the xPlore server:

1. Set up xPlore to support xQuery, using the index. See more details in the *EMC Documentum xPlore Installation Guide* and *EMC Documentum xPlore Administration Guide*.
2. Allow communication between the xPlore server and FS2 server.

Installing the Adapter Bundle

To install the adapter bundle, extract the adapter archive at the root level of the Federated Search Services installation. For example, if Federated Search Services is installed under `C:\Documentum\fs2`, extract the archive to this folder.

Unzipping `FS2_Adapter_core_Xplore.zip` installs the following components:

- The installation guide in `<FS2 installation directory>\docs\`
- `Xplore.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\core\Xplorebackends\`
- `core_Xplore.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
- Libraries in `<FS2 installation directory>\lib\`

After you complete the installation, you must create and configure a backend using the Admin Center, as described in the *Adapter Properties* section.

Adapter Properties

This section describes how to create a backend for the xPlore adapter. This section also provides details about the configurable properties for the xPlore adapter.

Creating a Backend

Complete the following steps to configure a backend for the xPlore adapter:

1. Log in to the Federated Search Services Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then click **Create new information source**.
4. Enter a name for the backend and select **Xplore** in the Intranet source list.
5. Follow the configuration wizard to set up the backend. The [Configurable Properties, page 544](#) provides information about the configurable properties of the backend.

When you create a backend, a new entry is added for your adapter in `www\docs\conf\domains.conf`. The *EMC Documentum Federated Search Services Administration Guide* provides more information about the Admin Center.

Configurable Properties

The xPlore adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the xPlore adapter.

bundle

Item	Description
ID	bundle
Description	Name and path of the adapter bundle used for a backend. The path is specified from the adapter repository (<code>docs\adapters</code>).
Default value	<code>core\Xplore.jar</code>

host

Item	Description
ID	host
Description	Host of xPlore server
Default value	none

client

Item	Description
ID	client
Description	Identifier sent to xPlore server to identify origin of queries
Default value	FS2

protocol

Item	Description
ID	protocol
Description	Protocol of xPlore server (http or https)
Default value	http

port

Item	Description
ID	port
Description	xPlore server port
Default value	none

queryLocale

Item	Description
ID	queryLocale
Description	Query locale required for language processing
Default value	en

domain

Item	Description
ID	domain
Description	xPlore domain, query scope
Default value	none

rootPath

Item	Description
ID	rootPath
Description	XPath of element to search in xPlore
Default value	none

timeout

Item	Description
ID	timeout
Description	Timeout for xPlore, in seconds
Default value	60

stopLimit

Item	Description
ID	stopLimit
Description	Maximum hits in xPlore
Default value	50

compoundScore

Item	Description
ID	compoundScore
Description	By default false, means not to compute relevance score from FS2 side again. See <i>EMC Documentum Platform and Platform Extensions Installation Guide</i> .
Default value	false

query

Item	Description
ID	query
Description	Attribute list used to query. For example, <code>query=full-text,title,author,date</code> .
Default value	none

result

Item	Description
ID	result
Description	Attribute list used to return results. For example, <code>result=title,author,date</code> .
Default value	none

Optional Properties

This section describes the optional properties for the xPlore adapter.

filter

Item	Description
ID	filter
Description	By default false, will not filter result set by FS2 server. See <i>EMC Documentum Platform and Platform Extensions Installation Guide</i> .
Default value	false

collection

Item	Description
ID	collection
Description	Collection in xPlore. Empty means all data collections under the domain. Comma separates multiple collections.
Default value	none

mapout.<FS2 attributes>

Item	Description
ID	mapout.<FS2 attributes>
Description	Specifies all XPath values of FS2 attribute in query and result list. For example: mapout.full-text= mapout.title=dmftmetadata//object_name mapout.date=dmftmetadata//r_creation_date mapout.author=dmftmetadata//owner_name mapout.modifier=dmftmetadata//r_modifier mapout.format=dmftmetadata//a_content_type mapout.size=dmftmetadata//r_full_content_size mapout.URL=dmftmetadata//r_object_id
Default value	mapout.full-text=.

Troubleshooting

If you encounter a connection or configuration issue, try the following tasks to solve the issue:

- Test querying Admin. Enable trace to get details.
- Verify query from xPlore server.
- Verify network connection between FS2 server and xPlore server.

InfoArchive adapter installation

Introduction

The EMC Documentum Federated Search Services InfoArchive adapter makes the EMC InfoArchive server searchable from any Federated Search Services client. This section provides instructions for installing the InfoArchive adapter. The *EMC Documentum Platform and Platform Extensions Installation Guide* provides more information about adapters, backend configuration, and common Federated Search Services attributes.

The most up-to-date information about Federated Search Services is available in the *EMC Documentum Platform and Platform Extensions Release Notes*.

This content is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing the Adapter

This section provides information on installation requirements and installation instructions.

Installation Requirements

The InfoArchive adapter supports EMC InfoArchive 3.1. It can also support other versions if the query web service is compatible.

The InfoArchive adapter is compatible with Federated Search Services 7.3.

Configuring InfoArchive Server

Complete the following steps to configure the InfoArchive adapter for the InfoArchive server:

1. Activate the web services on the InfoArchive server to allow communication with the adapter.
2. Submit a search from the InfoArchive site and make sure that the search returns results.
3. Make sure that the Query Web Services is available from the Federated Search Services server.

Installing the Adapter Bundle

To install the adapter bundle, extract the adapter archive at the root level of the Federated Search Services installation. For example, if Federated Search Services is installed under `C:\Documentum\fs2`, extract the archive to this folder.

Unzipping `FS2_Adapter_core_EIA.zip` installs the following components:

- The installation guide in `<FS2 installation directory>\docs\`
- `EIA.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\core\EIA\`
- `core_EIA.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
- Libraries in `<FS2 installation directory>\lib\`

After you complete the installation, you must create and configure a backend using the Admin Center, as described in [Adapter Properties, page 523](#).

Adapter Properties

This section describes how to create a backend for the InfoArchive adapter. This section also provides details about the configurable properties for the InfoArchive adapter.

Creating a Backend

Complete the following steps to configure a backend for the InfoArchive adapter:

1. Log in to the Federated Search Services Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then click **Create new information source**.
4. Enter a name for the backend and select **EIA** in the Intranet source list.
5. Follow the configuration wizard to set up the backend. The [Configurable Properties, page 544](#) provides information about the configurable properties of the backend.

When you create a backend, a new entry is added for your adapter in `www\docs\conf\domains.conf`. The *EMC Documentum Federated Search Services Administration Guide* provides more information about the Admin Center.

Configurable Properties

The InfoArchive adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory Properties

This section describes the mandatory properties for the InfoArchive adapter. Most of them are straightforward from Admin Console.

bundle

Item	Description
ID	bundle
Description	Name and path of the adapter bundle used for a backend. The path is specified from the adapter repository (<code>docs\adapters</code>).
Default value	<code>core\EIA.jar</code>

host

Item	Description
ID	host
Description	Host name of the InfoArchive service
Default value	None

hasSecurity

Item	Description
ID	hasSecurity
Description	Specifies whether authentication is needed.
Default value	false

loginName

Item	Description
ID	loginName
Description	If <code>hasSecurity</code> is true, <code>loginName</code> must be provided.
Default value	None

loginPassword

Item	Description
ID	loginPassword
Description	If <code>hasSecurity</code> is true, <code>loginPassword</code> must be provided
Default value	None

consumerApplication

Item	Description
ID	consumerApplication
Description	Application name that is allowed to search
Default value	None

roles

Item	Description
ID	roles
Description	InfoArchive roles. Multiple roles are separated by comma.
Default value	None

holdingName

Item	Description
ID	holdingName
Description	InfoArchive holding name
Default value	None

channelName

Item	Description
ID	channelName
Description	InfoArchive channel name
Default value	None

locale

Item	Description
ID	locale
Description	InfoArchive locale setting
Default value	en_US

resultSchema

Item	Description
ID	resultSchema
Description	InfoArchive result schema. Only the schema name is required as the parameter for InfoArchive.
Default value	None

query

Item	Description
ID	query
Description	Attribute list used to query. For example, <code>query=full-text,title,author,date</code> .
Default value	None

result

Item	Description
ID	result
Description	Attribute list used to return results. For example, <code>result=title,author,date</code> .
Default value	None

mapin

Item	Description
ID	mapin
Description	xPath with namespace of the elements in the returned results from InfoArchive. mapin maps to the attributes in the result list. Every attribute in the result list is mandatory. For example, mapin.title=urn:eas-samples:en:xsd:phonecalls.1.0:CustomerFirstName mapin.date=urn:eas-samples:en:xsd:phonecalls.1.0:SentToArchiveDate mapin.author=urn:eas-samples:en:xsd:phonecalls.1.0:CustomerID.
Default value	None

mapout

Item	Description
ID	mapout
Description	Maps the query attributes to InfoArchive attributes. For example, mapout.full-text=CustomerFirstName mapout.date=SentToArchiveDate.
Default value	None

Optional Properties

This section describes the optional properties for the InfoArchive adapter.

filter

Item	Description
ID	filter
Description	Specifies whether to filter results by the Federated Search Services server. If filter is set to true, for non-content-centric InfoArchive, the result has high possibility to be filtered out.
Default value	false

compoundScore

Item	Description
ID	compoundScore

Item	Description
Description	Specifies whether to re-calculate relevance score from Federated Search Services.
Default value	false

stopLimit

Item	Description
ID	stopLimit
Description	Maximum number of results from the InfoArchive server
Default value	50

dateFormat

Item	Description
ID	dateFormat
Description	Format of dates in InfoArchive
Default value	MM/dd/yyyy

Troubleshooting

If you encounter a connection or configuration issue, try the following tasks to solve the issue:

- Test querying Admin. Enable trace to get details.
- Verify the InfoArchive query web service from the InfoArchive server or application side.
- Verify the InfoArchive query web service from the Federated Search Services server.

Query Translation

This section describes how the Federated Search Services translator translates a query into an InfoArchive query.

Refer to the *EMC InfoArchive Development Guide* for more information about the InfoArchive query operators.

Federated Search Services Operator	InfoArchive Operator
CONTAINS	Contains
AFTER, EQUAL_GREATER	GreaterOrEqual

Federated Search Services Operator	InfoArchive Operator
BEFORE, EQUAL_LOWER	LessOrEqual
BETWEEN	LessOrEqual, GreaterOrEqual
EQUAL, EQUALS_TO	Equal
GREATER	Greater
LOWER	Less

Google Search Appliance adapter installation

Introduction

This section provides instructions for installing the EMC Documentum Federated Search Services adapter for Google Search Appliance. The Google Search Appliance adapter gets the content indexed by Google Search Appliance, which is searchable from Federated Search Services.

The most up-to-date information about Federated Search Services is available in the *EMC Documentum Federated Search Services Release Notes*.

The *EMC Documentum Federated Search Services Adapter Administration Guide* provides more information about adapters, backend configuration, and common Federated Search Services attributes.

This content is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing the Adapter

This section provides information on installation requirements and installation instructions.

Installation Requirements

There is no particular requirement to install this adapter. The search requests are simple HTTP requests to the Google Search Appliance server. The search results are expected to be in the HTML format. The adapter extracts results from the HTML result pages.

Installing the Adapter bundle

To install the adapter bundle, make sure to extract the adapter archive at the root level of your Federated Search Services installation. For example, if the Federated Search Services server is installed under `C:\Documentum\fs2`, extract the archive to this folder. If you are prompted to replace some files, accept the replacement.

Adapter Properties

This section lists the configurable properties of the Google Search Appliance adapter. You can set these properties during the creation of the backend in Admin Center. You can modify the backend properties whenever you need by editing them in Admin Center.

Mandatory Properties

This section describes the mandatory properties for the Google Search Appliance adapter.

bundle

Item	Description
ID	bundle
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter directory (<code>www/wrappers</code>).
Default value	<code>indexer/GSA.jar</code>

client

Item	Description
ID	client
Description	Any valid front end
Default value	none

host

Item	Description
ID	host
Description	Host name of the GSA search interface
Default value	none

output

Item	Description
ID	output
Description	Format of the search results. This value is set by default and should not be changed.
Default value	xml_no_dtd

proxystylesheet

Item	Description
ID	proxystylesheet
Description	Custom HTML results through application of the XSL stylesheet associated with the specified front end name (any valid front end name as mentioned in client property).
Default value	none

site

Item	Description
ID	site
Description	Name of a collection. Note that you can search over multiple collections using the property escaped 'OR' (pipe character) to separate collection names.
Default value	none (for example, site=collection1 collection2)

Optional Properties

This section describes the optional properties for the Google Search Appliance adapter.

client.overview

Item	Description
ID	client.overview
Description	A comma separated list of attributes that constitute the overview field in the HTML interface.
Default value	none

image

Item	Description
ID	image
Description	Special logo used to represent results for the backend; if the value of this property is "XX" (e.g. image=XX), the logo should be stored in \www\tomcat\webapps\ao\data\sources\source-XX.gif.
Default value	google

port

Item	Description
ID	port
Description	Port of the GSA search interface
Default value	80

query

Item	Description
ID	query
Description	The list of primary Federated Search Services attributes available for querying this source
Default value	title, body, full-text,site, URL, document_type

querySuffix

Item	Description
ID	querySuffix
Description	Enables you to add an extra constraint to limit the search. You have to follow the Google syntax.
Default value	none (for example, inurl:pdf)

result

Item	Description
ID	result

Item	Description
Description	The list of known Federated Search Services attributes returned by this source
Default value	title, URL, body, document_type, size, site, source

stopLimit

Item	Description
ID	stopLimit
Description	The adapter limits its response to this number. Maximum number of results returned by this source for a query.
Default value	50

Sample Backends

If you are not a GSA administrator or if you create a backend on a website that uses GSA, you must extract the parameters from the URL generated after submitting a search.

For example, see the Food and Drug Administration website at <http://www.fda.gov/>.

When you submit a test search on the query form, a URL is generated similar to the following:

```
http://google2.fda.gov/search?q
=test+&numgm=0&site=FDAgov&client
=FDAgov&proxystylesheet=FDAgov&num
=50&sort=&output=xml_no_dtd&ie
=UTF-8&oe
=UTF-8&restrict
=&getfields=*&filter
=1
```

The Federated Search Services properties are as follows:

- host=google2.fda.gov
- action=/search
- site=FDAgov
- client=FDAgov
- proxystylesheet=FDAgov

Microsoft Exchange adapter installation

Introduction

The Microsoft Exchange adapter makes the Microsoft Exchange Server searchable from any Federated Search Services client such as Webtop or CenterStage. This section provides instructions for installing the Microsoft Exchange adapter. The *EMC Documentum Federated Search Services Adapter Administration Guide* provides more information about adapters, backend configuration, and common Federated Search Services attributes.

The most up-to-date information about Federated Search Services is available in the *EMC Documentum Federated Search Services Release Notes*.

This content is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing the Adapter

This section provides information on installation requirements and installation instructions.

Installation Requirements

The Federated Search Services server must be located in the trusted domain of Microsoft Exchange Server. The Microsoft Exchange adapter is compatible with Microsoft Exchange 2007, 2010, and 2013.

Configuring Microsoft Exchange Server

The Microsoft Exchange adapter searches the Microsoft Exchange Server through its Web Services, which are installed and activated by default with Microsoft Exchange. Ensure that you can reach `https://<host>/ews/Services.wsdl` from the Federated Search Services server.

Installing the Adapter Bundle

To install the adapter bundle, extract the adapter archive at the root level of the Federated Search Services installation. For example, if Federated Search Services is installed under `C:\Documentum\fs2`, extract the archive to this folder.

Unzipping `FS2_Adapter_core_MSExchange.zip` installs the following components:

- The installation guide in `<FS2 installation directory>\docs\`
- `MSExchange.jar` in `<FS2 installation directory>\www\wrappers\core\`
- Backends sample file in `<FS2 installation directory>\www\wrappers\core\MSExchangebackends\`
- Configuration template `core_MSExchange.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\WebBasedAdmin\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`
- Library for Microsoft Exchange web service in `<FS2 installation directory>\lib\`

After you complete the installation, you must create and configure a backend using the Admin Center, as described in [Adapter Properties, page 523](#).

Adapter Properties

This section describes how to create a backend for the Microsoft Exchange adapter. This section also provides details about the configurable properties for the Microsoft Exchange adapter.

Creating a Backend

Complete the following steps to configure a backend for the Microsoft Exchange adapter:

1. Log in to the Federated Search Services Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then click **Create new information source**.
4. Enter a name for the backend and select **MSExchange** in the Intranet source list.
5. Follow the configuration wizard to set up the backend. The [Configurable Properties, page 544](#) provides information about the configurable properties of the backend.

The *EMC Documentum Federated Search Services Administration Guide* provides more information about the Admin Center.

Configurable Properties

You can configure properties of the Microsoft Exchange adapter properties during the backend creation. This section describes the properties you can configure for the Microsoft Exchange adapter.

Mandatory Properties

This section describes the mandatory properties for the Microsoft Exchange adapter.

bundle

Item	Description
ID	bundle
Description	Name and path of the adapter bundle used for a backend. The path is specified from the adapter repository (www/adapters).
Default value	core\MSExchange.jar

host

Item	Description
ID	host
Description	Host name of the Microsoft Exchange Server CAS (Client Access Server)
Default value	None

Optional Properties

This section describes the optional properties for the Microsoft Exchange adapter.

stopLimit

Item	Description
ID	stopLimit
Description	Maximum number of results returned by an adapter after Federated Search Services filtering
Default value	50

mailboxFolder

Item	Description
ID	mailboxFolder

Item	Description
Description	The user folder to search. By default, the search goes to the Inbox folder. Possible values are Inbox, SentItems, Drafts, and DeletedItems. It does not work for any other folder or value.
Default value	Inbox

messageRendition

Item	Description
ID	messageRendition
Description	This property applies when the user requests to view a document. The document is returned in native (eml format, Outlook Express, Mime-type=message/rfc822) or HTML. The HTML rendition displays a list of the attachments, but does not provide access to the attached files.
Default value	native

displayAttachments

Item	Description
ID	displayAttachments
Description	Enables to display the list of the files attached to the message (attachments attribute). Extracting the attachment list of a message has high impacts on performance. provides some examples of display time displaying or not displaying attachments.
Default value	false

The following table shows the comparison of display times with or without attachments:

Number of messages	Displaying attachments	Not displaying attachments
50 (all with attachments)	12 sec. (4 msgs/sec)	5 sec. (10 msgs/sec)
50 (20 with attachments)	8 sec. (6 msgs/sec)	4 sec. (12 msgs/sec)
600 (all with attachments)	105 sec. (6 msgs/sec)	22 sec. (27 msgs/sec)
1500 (600 with attachments)	125 sec. (12 msgs/sec)	40 sec. (35 msgs/sec)

displayEmailAddress

Item	Description
ID	displayEmailAddress

Item	Description
Description	<p>Enables to display the email addresses of the author (from field) and recipients (to, cc, and bcc fields) in addition to their names.</p> <p>If set to false, it only displays the names, such as "John Doe; Jane Doe".</p> <p>If set to true, it displays both the names and the email addresses: "John Doe <john.doe@emc.com>; Jane Doe <jane.doe@emc.com>".</p>
Default value	false

resolveDistributionList

Item	Description
ID	resolveDistributionList
Description	<p>If the recipient is a distribution list, this property enables you to display all the members of the distribution list.</p> <p>If set to false, it only displays the name of the distribution list, such as "Human Resources <humanresources@emc.com>".</p> <p>If set to true, it displays both the names of the distribution list and all the names of the members of this list: "Human Resources <humanresources@emc.com>; John Doe <john.doe@emc.com>; Jane Doe <jane.doe@emc.com>".</p>
Default value	false

duplicateKey

Item	Description
ID	duplicateKey
Description	<p>Identifies duplicate results from a source. By default, when this attribute is absent, results are identified by the 'URL' attribute.</p> <p>This property is required by DFC when the adapter uses an embedded to retrieve results. It allows to identify results and avoid issues related to the cache expiration.</p>
Default value	record-id

Troubleshooting

Messages Cannot Be Imported

With Documentum 6.6 or earlier, the mime-type for msg objects is missing from the dm_format table. Federated Search Services unifies results based on the mime-type. It is not possible to open msg results because they are not assigned the correct Documentum type.

To fix this issue, run the following DQL query in Documentum Administrator:

```
UPDATE dm_format OBJECTS SET mime_type='application/vnd.ms-outlook' WHERE "name"='msg'
```

Query Translation

This section describes how the Federated Search Services translator translates a query into a Microsoft Exchange query.

Attributes

Searchable Attributes

The following table lists the Federated Search Services attributes that are searchable and their mapping with Microsoft Exchange attributes, if any.

Federated Search Services attribute	Exchange attribute (or field)	Example or description
full-text	body OR title	full-text:contains:test => (body=test* OR title=test*)
body		Message body
title	subject	Message subject
date	sent-date	Date the message was sent
received-date	received-date	Date the message was received
author	from	There is only one sender user name
recipient	to	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set
cc	cc	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set

Federated Search Services attribute	Exchange attribute (or field)	Example or description
bcc	bcc	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set
importance	importance	Possible values: high, normal, or low
sensitivity	sensitivity	Possible values: private, confidential, or normal
hasAttachments		Boolean attribute: true or false

Returned Attributes

The following table lists the attributes that are returned by Microsoft Exchange and their mapping with Federated Search Services attributes, if any.

Exchange attribute (or field)	Federated Search Services attribute	Example or description
record-id	record-id	ID of the Exchange message
subject	title	Message subject
sent-date	date	Date the message was sent
sent-time	sent-time	Time the message was sent
received-date	received-date	Date the message was received
received-time	received-time	Time the message was received
from	author	Sender's name
to	recipient	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set
cc	cc	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set
bcc	bcc	List of recipient names and email addresses depending on how the properties resolveDistributionList and displayEmailAddress are set
size	size	Message size in kilobytes
importance	importance	Message importance
sensitivity	sensitivity	Message sensitivity

Exchange attribute (or field)	Federated Search Services attribute	Example or description
hasAttachments	hasAttachments	Boolean to indicate if there are attachments with the message
attachmentCount	attachmentCount	Number of attachments

Operators

The operators AND, OR, and ANDNOT are supported. However, the operators may not work in some cases. For example, the author attribute can only have a single value, which means you cannot define a query such as author=johndoe AND author=janedoe.

Proximity operators are not supported.

Microsoft SharePoint adapter installation

Introduction

The EMC Documentum Federated Search Services Microsoft SharePoint adapter makes the Microsoft SharePoint Server searchable from any Federated Search Services client such as Webtop or CenterStage. This section provides instructions for installing the Microsoft SharePoint adapter. The *EMC Documentum Platform and Platform Extensions Installation Guide* provides more information about adapters, backend configuration, and common Federated Search Services attributes.

The most up-to-date information about the product is available in *EMC Documentum Federated Search Services Release Notes*.

This content is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing the Adapter

This section provides information on installation requirements and installation instructions.

Installation Requirements

The Microsoft SharePoint adapter supports Microsoft SharePoint Server 2010 and 2013. The Microsoft SharePoint adapter searches the Microsoft SharePoint Server through its Query Service web service that can be reached at `http://<host>/_vti_bin/search.asmx`. The documents are retrieved through the Copy web service (`http://<host>/_vti_bin/copy.asmx`).

The Microsoft SharePoint adapter is compatible with Federated Search Service 6.7 SP2 and 7.1.

The Microsoft SharePoint adapter supports both NTLM authentication and Kerberos authentication protocols.

Configuring Microsoft SharePoint Server

Complete the following steps to configure the adapter for Microsoft SharePoint Server:

1. Activate the web services on SharePoint Server to allow communication with the adapter.
2. Submit a search from the Sharepoint site and make sure that it returns results.
3. Make sure that the Search Web Services is available from the Federated Search Services server.
Open the following URL from a navigator:

`http://host:port/_vti_bin/search.asmx`

Installing the Adapter Bundle

To install the adapter bundle, extract the adapter archive at the root level of the Federated Search Services installation. For example, if Federated Search Services is installed under `C:\Documentum\fs2`, extract the archive to this folder.

Unzipping `FS2_Adapter_MSSharepoint.zip` installs the following components:

- This installation guide in `<FS2 installation directory>\docs\`
- `MSSharepoint.jar` in `<FS2 installation directory>\www\wrappers\core\`
- `login.conf` and `krb5.conf` in `<FS2 installation directory>\docs\`
- Backends sample files in `<FS2 installation directory>\www\wrappers\core\MSSharepointbackends\`
- `core_MSSharepoint.xml` in the following locations:
 - `<FS2 installation directory>\admin\webapps\AdminCenter\data\xml\template\`
 - `<FS2 installation directory>\admin\webapps\WebBaseAdmin\data\xml\template\`
- Libraries in `<FS2 installation directory>\lib\`

After you complete the installation, you must create and configure a backend using the Admin Center, as described in [Adapter Properties, page 523](#).

Testing Queries

To test queries, explore the scopes and managed properties of a given SharePoint server, you can use the following query tool: <http://fastforsharepoint.codeplex.com/>. It helps identifying possible values for the properties: scope and attributes. It also allows you to test the connection with the SharePoint server web services.

Adapter Properties

This section describes how to create a backend for the Microsoft SharePoint adapter. This section also provides details about the configurable properties for the Microsoft SharePoint adapter.

Creating a Backend

Complete the following steps to configure a backend for the Microsoft SharePoint adapter:

1. Log in to the Federated Search Services Admin Center.
2. Select **Information source configuration and organization** to navigate to the domains page.
3. Click **Add** and then **Create new information source**.
4. Enter a name for the backend and select **Sharepoint** in the Intranet source list.
5. Follow the configuration wizard to set up the backend. The [Configurable Properties, page 544](#) provides information about the configurable properties of the backend.

When you create a backend, a new entry is added for your adapter in `www\docs\conf\domains.conf`. The *EMC Documentum Federated Search Services Administration Guide* provides more information on the Admin Center.

Configurable Properties

The Microsoft SharePoint adapter properties can be configured during the backend creation. All properties, either mandatory or optional, are described in this section.

Mandatory properties

This section describes the mandatory properties for the Sharepoint adapter.

bundle

Item	Description
ID	bundle
Description	Name and path of the adapter bundle used for this backend. The path is specified from the adapter repository (docs/adapters).
Default value	core/MSSharepoint.jar

host

Item	Description
ID	host
Description	Microsoft SharePoint Server HTTP server name. It must be a valid DNS name, such as sharepointserver or sharepointserver.mydomain.com
Default value	none

port

Item	Description
ID	port
Description	Microsoft Sharepoint Server Application Pool port.
Default value	80

Optional Properties

This section describes the optional properties for the Sharepoint adapter.

action

Item	Description
ID	action
Description	The path to the Web Services. For example, /Site.
Default value	None

stopLimit

Item	Description
ID	stopLimit
Description	Federated Search Services limits its response to this number. By default, the limit is set to 50.
Default value	50

optimizedStopLimit

Item	Description
ID	optimizedStopLimit
Description	The value of the SharePoint hint. It must be higher than the stopLimit property.
Default value	70

protocol

Item	Description
ID	protocol
Description	Hypertext Transfer Protocol (http). Set https if the connections are secured (SSL or TLS).
Default value	http

supportsLogin

Item	Description
ID	supportsLogin

Item	Description
Description	When set to true, the SharePoint Server requires a login and a password. If a user does not enter a login and a password, the adapter uses the credentials that you have set in this backend configuration, if any.
Default value	true

loginName

Item	Description
ID	loginName
Description	<p>The default login name to use when authenticating with the source. This default login name is used for any SharePoint user that did not specify a personal one for this backend. The syntax is:</p> <pre><domain_name>/<user_name></pre> <p>Active only when the property supportsLogin is set to true.</p> <p>For NTLM authentication, domain_name can be its shortname. For Kerberos authentication, domain_name must be the full qualified domain name (FQDN).</p>
Default value	None

loginPassword

Item	Description
ID	loginPassword
Description	<p>The default login password to use when authenticating with the source. This default login password is used for any SharePoint user that did not specify a personal one for this backend.</p> <p>Active only when the property supportsLogin is set to true.</p>
Default value	None

authentication

Item	Description
ID	authentication
Description	Specifies the authentication mode: ntlm or kerberos
Default value	ntlm

kerberos.env.krb5.conf

Item	Description
ID	kerberos.env.krb5.conf
Description	Kerberos configuration file. Refer to the template file <code>krb5.conf</code> in the <code>\fs2\docs\</code> directory. You must specify your Kerberos configuration and copy the file to your path or the default path <code>\fs2\www\docs\conf\krb5.conf</code> , which corresponds to the default relative path.
Default value	<code>..\..\docs\conf\krb5.conf</code>

kerberos.env.login.config

Item	Description
ID	kerberos.env.login.config
Description	Java login configuration file. Refer to the template file <code>login.conf</code> in the <code>\fs2\docs\</code> directory. By default, you do not need to change the template file. Copy the file to your path or the default path <code>\fs2\www\docs\conf\login.conf</code> , which corresponds to the default relative path.
Default value	<code>..\..\docs\conf\login.conf</code>

proxySet

Item	Description
ID	proxySet
Description	Set to true to use the default HTTP proxy to access to source. When set to true, set proxyHost and proxyPort properties as well.
Default value	false

proxyHost

Item	Description
ID	proxyHost
Description	The host name of the proxy. Set it when proxySet is set to true.
Default value	None

proxyPort

Item	Description
ID	proxyPort
Description	The port of the proxy. Set it when proxySet is set to true.
Default value	None

dateFormat

Item	Description
ID	dateFormat
Description	Microsoft SharePoint Server date format, for example, 2004-12-12T16:52:10Z.
Default value	yyyy-MM-dd'T'hh:mm:ss

query

Item	Description
ID	query
Description	The list of primary Federated Search Services attributes available for querying this source.
Default value	title, full-text, date, author, modifier, format, abstract, size, URL

queryMethod

Item	Description
ID	queryMethod
Description	The version to use for the query syntax
Default value	Keyword. Another value is Keyword2007, which is the legacy query syntax.

result

Item	Description
ID	result
Description	The list of known Federated Search Services attributes returned by this source.
Default value	title, author, date, URL, abstract, size

trusted

Item	Description
ID	trusted
Description	No filtering will be done by Federated Search Services on this list of attributes. It is often used when the source sends only partial results.
Default value	author, full-text, URL, abstract

scope

Item	Description
ID	scope
Description	Comma-separated list of Sharepoint scopes to restrict the search. The 'All Sites' scope is created during SharePoint installation. Additional scopes can be created by Site collection administrators. For example, All Sites, People, CustomScope.
Default value	All Sites

queryExtra

Item	Description
ID	queryExtra
Description	Additional query clause for all queries (using SharePoint Keyword query language). Example: contenttype:DocEMC2
Default value	None

attributes

Item	Description
ID	attributes
Description	Sharepoint queryable attributes.
Default value	rank, title, size, filetype, path, write

language

Item	Description
ID	language

Item	Description
Description	The language of the query terms. To find a value for a language, check the list of culture names provided by Microsoft: http://msdn.microsoft.com/en-us/globalization/bb896001.aspx . By default, it is set to en-US . To search items in Chinese, set this property to the Chinese locale zh-CN .
Default value	en-US

mapin.<SharePoint attributes>

Item	Description
ID	mapin.<SharePoint attributes>
Description	Defines the translation of a SharePoint attribute, available in the results, in a Federated Search Services attribute. Syntax: <code>mapin.<SharePoint attribute>=<FS2 attribute></code> If you define custom attributes in SharePoint, add their mapping to the configuration.
Default value	mapin.author=author mapin.description=abstract mapin.editor=modifier mapin.path=path mapin.linkurl=URL mapin.rank=rank mapin.size=size mapin.title=title mapin.write=date mapin.filetype=format mapin.description=abstract mapin.sitename=site

mapout.<FS2 attributes>

Item	Description
ID	mapout.<FS2 attributes>

Item	Description
Description	<p>Defines the translation of an FS2 attribute in a SharePoint attribute for querying the source. Syntax:</p> <pre>mapout.<FS2 attribute>=<SharePoint attribute></pre> <p>If you define custom attributes in SharePoint, add their mapping to the configuration.</p>
Default values	<pre> mapout.title=title mapout.full-text=ALL mapout.size=size mapout.date=write mapout.format=filetype mapout.modifier=editor mapout.author=author mapout.abstract=description mapout.URL=path </pre>

createTraceFiles

Item	Description
ID	createTraceFiles
Description	Set to true to create trace files: soap message sent to SharePoint (MOSS_Query.xml), result returned by SharePoint (MOSS_result.xml), statement sent to the SharePoint system (MOSS_query.txt).
Default value	false

traceFilePath

Item	Description
ID	traceFilePath
Description	Specifies the directory where trace files are created.
Default value	c:\temp\

Troubleshooting

Internet Information Services Settings

You must make sure that the anonymous access is not allowed. Using the Internet Information Services (IIS) Manager, open the Properties of the SharePoint website. Select **Directory Security > Authentication Methods** and unselect **Enable anonymous access**.

Kerberos Authentication Settings

Use the query tool (<http://fastforsharepoint.codeplex.com/>) to check whether the SharePoint Query web service is configured correctly. If the tool can work, it indicates that the Query web service is under the control of Kerberos authentication.

Use the test tool with TRACE from the Admin Center to check the generated principal. The realm of the generated principal must be consistent with `krb5.conf` so that the correct Key Distribution Center (KDC) and Domain can be mapped.

Use KDC tracing tools like Windows Event Viewer to trace the interaction between Federated Search Services, KDC, and applications.

If you still have problems with Kerberos authentication after trying the previous methods, you can use any other tool to verify that Kerberos authentication with given principal can work on the Federated Search Services server before focusing on the adapter itself.

System.ArgumentNullException Error: misconfiguration

The following table details the System.ArgumentNullException error and how to fix the corresponding issue.

Item	Description
Problem	<p>You cannot get results back from the SharePoint server.</p> <p>First you must set the <code>createTraceFiles</code> and <code>traceFilePath</code> properties.</p> <p>You get a "System.ArgumentNullException" error or the following result in the trace file <code>MOSS_result.xml</code>:</p> <pre><ResponsePacket xmlns="urn:Microsoft.Search.Response"> <Response domain=""> <Status>ERROR_SERVER</Status> <DebugErrorMessage>System.ArgumentNullException< /DebugErrorMessage> </Response></ResponsePacket></pre>
Cause	<p>It means that the Federated Search Services server could not reach the SharePoint server.</p>
Resolution	<p>Make sure that the system hosts files are updated as described in Configuring Microsoft SharePoint Server, page 557.</p>

Out of Memory Error: configuration limitation

The following table provides information on the configuration limitation of the Microsoft SharePoint Server adapter.

Item	Description
Problem	While configuring the Microsoft SharePoint Server adapter, an <i>out of memory</i> error may appear.
Cause	The <i>out of memory</i> error occurs due to a memory leak of the adapter in the Admin Center.
Resolution	Do not click the test button several times (more than ten times).

Query error: host definition

The following table provides information on the host definition of the Microsoft SharePoint Server adapter.

Item	Description
Problem	When configuring the Microsoft SharePoint Server adapter in Admin Center, the following <i>Query</i> error appears: <pre>xtrim.adapter.definition.AdapterException: Adapter [tmpBackends/tmp] raised error of type [QUERY] with message [Query Syntax ErrorERROR_SERVER Check the host value : use the hostname instead of the IP.]</pre>
Cause	This error occurs because the host value is an IP address and the adapter requires the server name for SharePoint 2010.
Resolution	Change the host value to a host name instead of an IP address as described in the section host , page 544.

Network error: host invalid

The following table provides information on the host value of the Microsoft SharePoint Server adapter.

Item	Description
Problem	When configuring the Microsoft SharePoint Server adapter in Admin Center, the following <i>Network</i> error appears: <pre>xtrim.adapter.definition.AdapterException: Adapter [core/Sharepointbackends/Tamise6] raised error of type [NETWORK] with message [Communication failure. Could not reach the server. Check the host value.]</pre>
Cause	This communication error occurs when the host value is invalid or if the Microsoft SharePoint Server cannot be reached.
Resolution	Make sure that the host value is valid and the Microsoft SharePoint Server is running.

Login error: credentials invalid

The following table provides information on the loginName and loginPassword values of the Microsoft SharePoint Server adapter.

Item	Description
Problem	When configuring the Microsoft SharePoint Server adapter in Admin Center, the following <i>Login</i> error appears: <pre>xtrim.adapter.definition.AdapterException: Adapter [tmpBackends/tmp] raised error of type [LOGIN] with message [Check your login and your password information. Please note that a Windows domain name is mandatory.]</pre>
Cause	This error occurs when the supportsLogin property is set to true and the loginName and/or loginPassword properties are not defined.
Resolution	Set the value of the loginName and loginPassword properties as described in the sections loginName, page 546 and loginPassword, page 546 .

Query Translation

The Federated Search Services translator translates a query into Microsoft SharePoint Keyword Query Language. This section describes how the queries are translated.

Refer to the Microsoft SharePoint Keyword Query Language documentation for more information.

Constraints operators

The supported constraint operators are as follows:

AND

Item	Description
Description	Use AND when all listed terms must be included.
FS2 operator	AND
SharePoint operator	AND
Example	("meeting*" AND "brainstorm*")

OR

Item	Description
Description	Use OR to include either of the listed terms.

Item	Description
FS2 operator	OR
SharePoint operator	OR
Example	("meeting*" OR "brainstorm*")

ANDNOT

Item	Description
Description	Use ANDNOT to exclude a term.
FS2 operator	ANDNOT
SharePoint operator	NOT
Example	("meeting*" NOT "brainstorm*")

NEAR

Item	Description
Description	Use NEAR to define the proximity between two terms.
FS2 operator	NEAR
SharePoint operator	NEAR
Example	("meeting*" NEAR(10) "brainstorm*")

Wildcard

Item	Description
Description	Use a wildcard to replace 0 to more characters.
FS2 operator	*
SharePoint operator	*
Example	meet*

Operators

The supported operators are listed in the following table:

Name	Type	Example
AFTER	Date	"write">"2012-02-28"
BEFORE	Date	"write"<"2012-02-28"

Name	Type	Example
CONTAINS	String	"test*"
EQUAL	Numerical	"size"="153600"
EQUAL_GREATER	Numerical	"size">="153600"
EQUAL_LOWER	Numerical	"size"<="153600"
EQUALS_TO	String	"author": "mickael"
GREATER	Numerical	"size">"153600"
LOWER	Numerical	"size"<"153600"

Twitter adapter installation

Introduction

This section provides instructions for installing the EMC Documentum Federated Search Services Twitter adapter. The Twitter adapter queries tweets from `www.twitter.com`.

The most up-to-date information about Federated Search Services is available in the *EMC Documentum Federated Search Services Release Notes*.

The **Documentum Federated Search Services Adapter** chapter provides more information about adapters, backend configuration, and common Federated Search Services attributes.

This content is intended for Federated Search Services administrators and librarians:

- The administrator configures the Federated Search Services server. The administrator is responsible for the technical configuration of the system, including the definition of backends.
- The librarian works in cooperation with the administrator to organize the backends into domains that make most sense to the end users.

Installing the Adapter

This section provides information on installation requirements and installation instructions.

Installation Requirements

The Microsoft SharePoint adapter supports Microsoft SharePoint Server 2010 and 2013. The Microsoft SharePoint adapter searches the Microsoft SharePoint Server through its Query Service web service that can be reached at `http://<host>/_vti_bin/search.asmx`. The documents are retrieved through the Copy web service (`http://<host>/_vti_bin/copy.asmx`).

The Microsoft SharePoint adapter is compatible with Federated Search Service 6.7 SP2 and 7.1.

The Microsoft SharePoint adapter supports both NTLM authentication and Kerberos authentication protocols.

Configuring Microsoft SharePoint Server

Complete the following steps to configure the adapter for Microsoft SharePoint Server:

1. Activate the web services on SharePoint Server to allow communication with the adapter.
2. Submit a search from the Sharepoint site and make sure that it returns results.
3. Make sure that the Search Web Services is available from the Federated Search Services server. Open the following URL from a navigator:

`http://host:port/_vti_bin/search.asmx`

Installing the Adapter Bundle

To install the adapter bundle, make sure to extract the adapter archive at the root level of your Federated Search Services installation. For example, if your Federated Search Services server is installed under `C:\Documentum\fs2`, extract the archive to this folder. If you are prompted to replace some files, accept the replacement.

Preparing Twitter Account

To use the Twitter adapter, you must register an account on `twitter.com` and apply an application access ticket for the registered account. The application access ticket contains the information you need for the `oauth.consumerKey` and `oauth.consumerSecret` properties, which are described in .

Adapter Properties

This section lists the configurable properties of the Twitter adapter. You can set these properties during the creation of the backend in Admin Center. You can modify the backend properties whenever you need by editing them in Admin Center.

The *EMC Documentum Federated Search Services Administration Guide* provides more information about the Admin Center.

Mandatory Properties

This section describes the mandatory properties for the Twitter adapter.

oauth.consumerKey

Item	Description
ID	oauth.consumerKey
Description	Consumer key token used to access public API you have applied
Default value	None

oauth.consumerSecret

Item	Description
ID	oauth.consumerSecret
Description	Consumer secret token used to access public API you have applied
Default value	None

Optional Properties

This section describes the optional properties for the Twitter adapter.

language

Item	Description
ID	language
Description	The language to query. This property supports all the languages that <code>twitter.com</code> supports.
Default value	en

compoundScore

Item	Description
ID	compoundScore

Item	Description
Description	Set to true to re-calculate score and re-rank the results based on information from the source and query.
Default value	true

REST Services

This chapter is intended primarily for administrators who are installing Documentum Platform REST Services.

Introduction

EMC Documentum Platform REST Services is a set of RESTful web service interfaces that interact with the Documentum Platform.

DFC Configuration

The `dfc.properties` file provides property settings for the Documentum Foundation Classes runtime. Documentum Platform REST Services depends on these property settings. The `dfc.properties` file is located in `WEB-INF/classes` when you deploy the WAR file.

You can use a `#include` statement to point to a properties file outside of the web application on the local file system. This operation can make access to some settings more convenient and allows you to modularize your configuration settings:

```
#include C:\Documentum\config\dfc.properties
```

Docbroker and Global Registry Properties

The `dfc.properties` file includes critical settings that are required for RESTful Services to reach a connection broker (also called a docbroker) and connect to Content Server.

Property	Value
<code>dfc.docbroker.host[0]</code>	The fully qualified hostname for the connection broker. You can add backup hosts by adding new properties and incrementing the index number within the brackets.
<code>dfc.docbroker.port[0]</code>	When you use a port for the connection broker other than the default of 1489, add a port key.

Property	Value
<code>dfc.globalregistry.repository</code>	The global registry repository name.
<code>dfc.globalregistry.username</code>	The username of the global registry user. The global registry user, who has the default username <code>dm_bof_registry</code> , must have read access only to the objects that are in the <code>/System/Modules</code> directory and the <code>/System/NetworkLocations</code> directory.
<code>dfc.globalregistry.password</code>	An encrypted password value for the global registry user.

You have one of the following options, either:

- Copy the username and encrypted password for the global registry user from the `dfc.properties` file on the global registry Content Server host
- Select another global registry user and encrypt the password using the following command:

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

Apache Tomcat

Follow these steps to perform a simple WAR file deployment on an Apache Tomcat server:

1. Stop the Apache Tomcat server.
2. Copy the WAR file to the `TomcatHome/webapps` directory.
3. Start the Apache Tomcat server.

VMware vFabric tc Server

The following documentation provides detailed information about how to deploy a web application on the vFabric tc Server: [Deploy a Web Application to a tc Server Runtime Instance](#).

Oracle WebLogic Server

You must turn off HTTP Basic authentication on the container before deploying Documentum Platform REST Services on Oracle WebLogic.

To turn off HTTP Basic authentication:

1. Edit the following file and save it to the following location: `<WebLogic_Home>/user_projects/domains/<Domain>/config/config.xml`.

- a. Find the `<security-configuration>` section of the file.
- b. If `enforce-valid-basic-auth-credentials` is already defined in this section, change its value to `false`. Otherwise, add the following line before the `</security-configuration>` line:


```
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
```

2. Restart the WebLogic server.

After you turn off HTTP Basic authentication, deploy the WAR file for RESTful Services using the WebLogic Console.

For detailed documentation on how to deploy a web application on WebLogic, see [Understanding the Deployment Process](#).

When you deploy RESTful Services on WebLogic 12c running JDK7 in Kerberos authentication mode, you must specify the location of the JAAS configuration file on WebLogic 12c by using a JVM parameter.

For example, you can use the following JVM parameter:

```
-Djava.security.auth.login.config=<Path>/jaas.config
```

Where the value of `<Path>` is typically one of the following:

- `<Your_Weblogic_Server_Root>/User_Projects/Domains/Mydomain/Bin/SETDOMAINENV.BAT`
- `<Your_Weblogic_Server_Root>/User_Projects/Domains/Mydomain/Bin/SETDOMAINENV.SH`



Caution: Specifying the location of the JAAS configuration file in the `rest-api-runtime.properties` file does not work.

IBM WebSphere

Follow these steps to deploy Documentum Platform REST Services on IBM WebSphere using the Integrated Solutions console:

1. Start WebSphere server.
2. Add a custom property:
 - a. Select **Application servers > ServerName > Web container > Custom properties**.
 - b. Add a custom property:


```
com.ibm.ws.webcontainer.remove trailing servlet path slash
```

 Set its value to `true`.
3. Install the WAR file.
4. Configure the class loader policy for Documentum Platform REST Services by setting the **Class Loader Order** property to **"Classes loaded with local class loader first (parent last)"**.
5. Start Documentum Platform REST Services.

Red Hat JBoss EAP

Deploy Documentum Platform REST Services on Red Hat JBoss EAP:

JBOSS EAP 6 and later are different from the previous versions of JBOSS EAP 5 or 4. You can deploy Documentum Platform REST Services on JBOSS EAP 6 and later in **Kerberos authentication in Standalone Mode** or in **Domain Mode**.

- **Kerberos authentication in Standalone Mode:**

To deploy Documentum Platform REST Services in the standalone mode, copy the war file to the `<JBoss_Installation_Dir>/standalone/deployments` directory.

The JAAS configuration that is defined in the `krbJaas.conf` file that comes by default in the JBOSS Server, is not loaded for security reasons.

When Documentum Platform REST Services is deployed using the Kerberos authentication standalone mode, the workaround for the above limitation is to add your JAAS configuration, as XML code, into the `/jboss-eap-6.4/standalone/configuration/standalone.xml` file.

Here is a code sample that shows you how to add a JAAS configuration using XML code:

Example 9-1. JBOSS JAAS Configuration

```
<security-domain name="HTTP-REST" cache-type="default">
  <authentication>
    <login-module
      code="com.dstc.security.kerberos.jaas.KerberosLoginModule"
      flag="required">
      <module-option name="debug" value="true"/>
      <module-option name="principal" value="HTTP/REST"/>
      <module-option name="realm" value="SUN.PLANET.COM "/>
      <module-option name="refreshKrb5Config" value="true"/>
      <module-option name="noTGT" value="true"/>
      <module-option name="useKeyTab" value="true"/>
      <module-option name="storeKey" value="true"/>
      <module-option name="doNotPrompt" value="true"/>
      <module-option name="useTicketCache" value="false"/>
      <module-option name="isInitiator" value="false"/>
      <module-option name="keyTab" value="/root/tim.SUN.keytab"/>
    </login-module>
  </authentication>
</security-domain>
```

- **Domain Mode:** For more information on how you can deploy Documentum Platform REST Services in Domain mode, see [Deploy an Application in a Managed Domain Using the Management CLI](#).

Deploy and Configure REST Services on a Docker Environment

1. Install the supported version of Docker in your host machine.
2. Prepare your configuration files and put them, as needed, in one or both of the following locations:
`<CURRENT_DIR>/rest/config/dfc.properties`
`<CURRENT_DIR>/rest/config/rest-api-runtime.properties`
3. Run the following command in the current directory:

```
docker run --name rest -p 8080:8080 -d -v `pwd`/config:/root/rest  
/config -v `pwd`/logs:/root/rest/logs <Rest_Image_Name>
```
4. To verify the installation, try the following URL:
`http://localhost:8080/dctm-rest/services`
If you can access the above URL, then your installation of Documentum Platform REST Services on a Docker environment is successful.

Validating REST Deployment

To validate that your RESTful Services deployment is successful, access the Home Document resource in your web browser. The Home Document resource provides an entry point to all of the other available resources.

Validate a REST Services Instance on a Localhost

1. To validate that a RESTful Services instance is successfully deployed on localhost at port 8080, try the following URL:
`http://localhost:8080/dctm-rest/services`
If you can access the above URL, then your installation of Documentum Platform REST Services is successful.

Thumbnail Server

This chapter describes the product and how it works to serve thumbnails to the Documentum clients.

Thumbnail Server retrieves low-resolution image files from a designated thumbnail file store on the Content Server host. These images are then used as visual cues in browsers and other Web client applications.

Two types of thumbnails are served by Thumbnail Server:

- Preview images created by Content Transformation Services server when an applicable file type is imported or checked in to the repository. These thumbnails are saved in the thumbnail store on the Content Server.
- Default thumbnail images per type or format. These are icons that can be specified for folders or unknown file types. These thumbnails are stored in the repository, and can be viewed or updated when necessary.

Thumbnail Server uses Java servlets to manage thumbnail representations, and HTTP technology to accelerate the display of thumbnail images in Web client applications. The file store is designated as a thumbnail file store when its `media_type` attribute is set to 1.

Applications retrieve thumbnails from thumbnail file stores by constructing a URL for the file. The URL consists of a main servlet which requires a path, store, and ticket argument, if applicable.

Introduction

All requests for the system to return thumbnails are processed by the Thumbnail Server. The following sections describe how the Thumbnail Server works when a thumbnail is requested from an application.

Thumbnails and Thumbnail Server

A thumbnail is an image that is used to visually represent a file in client applications. Thumbnail renditions provide a visual cue for browsing files, and they enable users to quickly identify objects at a glance, since the object name and/or additional attributes may not always provide enough information about an object.

Upon registration of an object, the Content Transformation Services server automatically generates thumbnail renditions of the object by passing it to the appropriate plug-in. The plug-in extracts

the object's media properties and creates a new object by transforming the original object into a predefined thumbnail format. The Content Transformation Services server sends the thumbnail back to Content Server as a rendition of the original object and saves the thumbnail's properties as attributes of that rendition. Content Server stores thumbnails in a special file store that is shared with the Thumbnail Server.

The Thumbnail Server is a separate server (which must be downloaded and installed on the Content Server host) that interacts directly with a client browser. The Thumbnail Server runs on a servlet engine, which itself runs on an HTTP server. This means that Java servlets are used to manage thumbnail representations and HTTP or HTTPS protocol is used to communicate with the web client applications.

For Documentum configurations that require SSL security, a special configuration is required to run Thumbnail Server with an HTTPS protocol. [Configuring Thumbnail Server for SSL, page 578](#) provides more information to configure the Thumbnail Server for SSL.

Requesting thumbnails

When an application requests a thumbnail file, it sends the request to the Content Server which sends the thumbnail URL back to the application. The application uses the thumbnail URL in a browser page. When the page is about to display, the client (browser) will contact Thumbnail Server (using the same URL) to obtain the desired thumbnail rendition.

The Thumbnail Server consists of a main servlet which requires a path, store, and ticket argument, if applicable, in the URL.

- The path argument is the path of the thumbnail on the Content Server, relative to the root location of the thumbnail storage area; for example, 00232803/80/00/01/0b.jpg.
- The store argument is the name of the storage area where the file is stored, or the distributed store name if stored in a distributed store. For example, the store can be called thumbnail_store_01.
- If ticket security has been enabled, the third argument will be a ticket argument, which is a string that encrypts the above parameters and a time stamp. For example, the ticket may be 76WR4QJ89X102.

With the above examples, a URL to request a thumbnail may appear as follows:

```
http://MyContentServer:8081/thumbsrv/getThumbnail?  
path=00232803\80\00\01\0b.jpg&store=thumbnail_store_01&  
ticket=76WR4QJ89X102&format=jpeg_th&page_modifier=medium_jpeg_th&  
did=090004d280001c70
```

When the servlet is invoked, it extracts the required URL arguments. If the URL contains a ticket, the Thumbnail Server decrypts it and matches it against the first two arguments of the URL (for example, path=00232803/80/00/01/0b.jpg&store=thumbnail_store_01&ticket=76WR4QJ89X102). If it matches, the time stamp is checked. If the time stamp is less than the current time, then the request is rejected. This means that the five minute time limit has expired. [Changing the ticket time-out value, page 584](#) contains instructions for changing the default ticket time-out.

After the URL has passed the security check, the servlet determines the repository where the thumbnail is stored. This is determined by the path argument which is the repository ID. The first time a request for a particular repository is made, the servlet will connect to it and retrieve the storage area with the specified name. Next, it checks whether it is really a thumbnail storage area

(attribute `media_type=1`). If not, an error is returned. Next, it checks if the storage area requires a ticket. If it does, and no ticket has been provided in the URL, the request is rejected. Otherwise, the root path of the storage area is retrieved and cached for subsequent requests. The complete path to the thumbnail is then constructed by concatenating the two paths. Finally, the thumbnail file is returned to the browser.

If the request fails any of the tests detailed above, an HTTP 400 error (Bad Request) is returned to the client's browser. However, a detailed error message will be written to the Thumbnail Server log file, `localhost.xxxx-xx-xx.log`.

If either the path or store argument is missing from the URL received by the server, the server uses the arguments that are present to choose a default thumbnail to return to the application. [Serving default thumbnails, page 569](#) describes how default thumbnails are served to the browser.

Serving default thumbnails

A *default thumbnail* is a thumbnail file used as a substitute for the requested thumbnail file, for documents that do not have an associated preview image. The thumbnail to be served can be based on three characteristics of a document: object type, format, and whether it is a virtual document. A default thumbnail will usually appear as a generic icon representing the object's file type.

Sometimes an application requests a URL to a non-existent thumbnail rendition of an object. For example, when you request a thumbnail for an object that was just deleted by another user, or if the object was just registered, and Media Transformation Server or Documents Transformation Server has not yet had a chance to generate a thumbnail. When that occurs, the application can use the information returned by Content Server (by the `THUMBNAIL_URL` keyword or the `GET_FILE_URL` administrative method, depending on which security method you choose) to build a URL that will return a default thumbnail. [Thumbnailing, page 569](#) provides more information on security when requesting thumbnails.

Default thumbnails are stored on the repositories configured for Thumbnail Server. The Thumbnail Server determines which thumbnail to serve based on rules defined in the `default_thumbnails.xml` file. The `default_thumbnails.xml` file's elements describe a list of rules for matching parameters in a URL with a default thumbnail. The `default_thumbnails` rules file conforms to the `default_thumbnails.dtd`.

[Understanding default thumbnails, page 583](#) provides instructions to obtain a default thumbnail. [Adding default thumbnails, page 584](#) provides instructions to add a default thumbnail. [Changing the ticket time-out value, page 584](#) provides instructions to change the encryption key.

Installation Overview

Thumbnail Server serves graphic files from a thumbnail file store on the Content Server host. It provides these thumbnails to rich media-enabled client applications. Thumbnails are renditions created by the Content Transformation Services server when a file is imported or checked in to the repository.

Installing Thumbnail Server installs the Thumbnail Server software and sets the `base_url` attribute for file store objects that are defined for repositories residing on the host thumbnail storage areas.

The `base_url` is set to a URL that identifies the Thumbnail Server. Its value is used by applications to construct complete URLs for files in the thumbnail storage area.

A file store is a thumbnail storage area when its `media_type` attribute is set to 1.

On Windows hosts, the Thumbnail Server is installed as a service, set to start automatically upon system restart.



Caution: If you are upgrading Content Server from a release earlier than 7.3, then uninstall the previous version of Thumbnail Server first. Install and configure Thumbnail Server 7.3 after the Content Server 7.3 upgrade is complete.

Pre-Installation Requirements and Tasks

Thumbnail Server must be installed on the same host as your Documentum Content Server. The Thumbnail Server requires at least 32.5 MB of disk space.

EMC Documentum Platform and Platform Extensions Release Notes provides the information on hardware, operating system requirements, and software version requirements.

Setup verification

Prior to installation, verify the following:

- WinZip or a similar file-extraction utility is installed on the host machine(s).
- If you already have a `dfc.properties` file, ensure that it is pointing to the correct Connection Broker. [Identifying a Connection Broker in `dfc.properties`, page 570](#) provides the instructions.
- If the docbase is lockbox-enabled, then the `dfc.properties` file should contain the crypto repository details:

```
dfc.crypto.repository=<docbase_name>
```
- The Connection Broker and repository services are running on your Content Server. [Checking Connection Broker and repository services, page 571](#) provides more information.
- The Content Server is installed and a repository has been created.

Identifying a Connection Broker in `dfc.properties`

If you already have Documentum Foundation Classes (DFC) installed on your Thumbnail Server host, you will have a `dfc.properties` file. Follow the procedure below to ensure that the correct Connection Broker is identified. If you are installing Thumbnail Server on a clean host you will not have a `dfc.properties` prior to installation, and this procedure is not required.

To identify the correct Connection Broker in `dfc.properties`:

1. Search for the `dfc.properties` file on your host. The file is usually located in the `$DOCUMENTUM/config` folder.

2. Open `dfc.properties` in a text editor.
3. Find the line indicating the Connection Broker. Ensure that the specified Connection Broker is the one that connects to the repository. Change it if necessary.
4. Save and close the `dfc.properties` file.

Checking Connection Broker and repository services

The Connection Broker and repository services must be of version 6 or later, and running properly on the Content Server host before you install Thumbnail Server.

Downloading installer

Before beginning the installation process, it is best to have the installer ready and available on your Thumbnail Server host. The Thumbnail Server installer can be found on EMC Online Support

<https://support.emc.com>.

Installing, configuring, and uninstalling Thumbnail Server

This section explains how to install, configure, unconfigure, and uninstall Thumbnail Server:

Required Thumbnail Server installation information

This section provides a table for recording the information you need to install Thumbnail Server. Having this information ready and available prior to installation ensures the accuracy and efficiency of your installation.

Installation information	Description
The host computer's Windows domain or the machine name	If you are installing Thumbnail Server on a Windows host, you must know the host's Windows domain. If the host is not part of a Windows domain, you must know the machine name.
Password for the installation owner	This is the same password as the one used to log in to the host.

Installation information	Description
Thumbnail Server and administration port numbers	<p>These ports are used by the Thumbnail Server and its administration tool.</p> <p>The default Thumbnail Server port is 8081.</p> <p>The default administration port is 8008.</p> <p>Leave these ports as default whenever possible.</p>
User names, passwords, and AEK passphrase (if it is not the default passphrase) for repositories you want to configure with Thumbnail Server	For each repository you choose to configure with the Thumbnail Server, a SuperUser name and password is required.

Installing and configuring Thumbnail Server on Windows

Use these instructions to install Thumbnail Server on a Windows host. The installer installs the Thumbnail Server software. All running repositories on the host can be configured by setting the `base_url` for all file store objects that are also thumbnail storage areas.

You can rerun the Configurator at any time to configure additional repositories.

To install Thumbnail Server on a Windows host:

1. Log in to the Content Server host as the Documentum Content Server installation owner.
2. Remove any previous versions of Thumbnail Server using the instructions in [Uninstalling Thumbnail Server from a Windows host, page 580](#).
3. Recall where the installation files are located.
4. Extract the Thumbnail Server installer bundle to a temporary location.
5. Double-click the file `thumbserverSetup.exe` to launch the installer.
The Thumbnail Server Installer splash screen is displayed, followed by the Welcome screen.
6. Click **Next**.
The license agreement terms are displayed.
7. Select the option to agree with the terms of the license agreement, then click **Next**.
If you do not have the correct DFC installed on the Content Server host, the installer will fail at this point.
You are prompted to select if the Thumbnail Server must run in the HTTP mode or HTTPS (SSL) mode.
8. Select the appropriate option, then click **Next**.
Note: If you have chosen to run in HTTPS mode, then you must also configure Thumbnail Server for SSL. [Configuring Thumbnail Server for SSL, page 578](#) provides the instructions.
9. Enter the number of a port that is available for Thumbnail Server or click **Next** to accept the default port (8081).
If you enter a port number already in use, you will be prompted to enter another port number.

10. Enter the number of a port that is available for Thumbnail Server administration or click **Next** to accept the default port (8008).

The installation summary screen appears, indicating the applications that will be installed.

- If you are satisfied with the installation confirmation, click **Next**.
- If you want to change a component of the installation, use the Back button to navigate back through the installer and change the install information.

The installation proceeds, installing the Thumbnail Server and its Configurator. The installer creates the following folders in the directory %DM_HOME%\thumbsrv\:

- \conf
- \configurator
- \container
- \install
- \logs

The post-installation notification screen appears, informing you to run the Thumbnail Server Configurator to add support to serviceable repositories.

11. Make note of the Configurator location and click **Next**.

A message notifies you that Thumbnail Server was installed successfully.

12. Click **Done** to close the installer.

To configure repositories for Thumbnail Server on a Windows host:

1. Navigate to the Thumbnail Server Configurator under **Start > Programs > Documentum > Apply Thumbnail Server support to docbase**.

The Thumbnail Server Configurator splash screen is displayed, followed by the Welcome screen.

Note: Your Connection Broker and any repositories requiring Thumbnail Server must be running to use the Configurator.

2. Click **Next**.

The license agreement terms are displayed.

3. Select the option to agree with the terms of the license agreement, and click **Next**.

4. The installer searches for all repositories configured for your Content Server host (in the folder %DOCUMENTUM%\dba\config\). Select a repository from the list and click **Next**.

You are prompted to enter the name of a repository SuperUser, their password, and the domain (optional). The Thumbnail Server will use this information to communicate with the repository.

5. Enter the required information and click **Next**.

If the default AEK passphrase is used in the docbase configuration, then Thumbnail Server Configurator will automatically use the default AEK passphrase.

If the docbase is configured with the custom AEK passphrase, then the Thumbnail Server Configurator displays a screen for the user to enter the AEK passphrase.

6. Enter the AEK passphrase.

If the repository you have chosen is part of a Distributed Content Server environment, a list of server config objects will appear for the repository.

7. Select a server config object from the list and click **Next**.

A message notifies you that the repository was successfully configured for Thumbnail Server.

8. Click **Done** to close the Configurator. The Thumbnail Server is started automatically.

Rerun the Configurator for any other repositories you wish to configure for the Thumbnail Server.

If you create a new repository on a host where Thumbnail Server is installed, you can run the Thumbnail Server Configurator to search for new repositories and configure them automatically.

Installing and configuring Thumbnail Server on non-Windows

Use these instructions to install Thumbnail Server on a non-Windows Content Server and configure it for any repositories served by the Connection Broker(s) listed in your `dfc.properties` file. For these instructions, when we refer to non-Windows, we also mean the other supported UNIX and Linux variations. Repositories are configured for the Thumbnail Server by setting `base_url` for the file store objects associated with thumbnail storage areas.

You can rerun the Thumbnail Server Configurator at a later time to configure additional repositories.

To install Thumbnail Server on non-Windows:

1. Log in to the Content Server host as the Documentum Content Server installation owner.
2. Remove any previous versions of Thumbnail Server using the instructions in [Uninstalling Thumbnail Server from a non-Windows host](#), page 580.
3. With repository and docbroker running, add the following classpath where “`$DOCUMENTUM_SHARED`” is the path that contains the `dctm.jar` and config folder:
`CLASSPATH=$DOCUMENTUM_SHARED/dctm.jar:$DOCUMENTUM_SHARED/config:$CLASSPATH`
4. Recall where the installation files are located.
5. Use the tar utility to transfer the downloaded file to a temporary directory using the following command applicable for your operating system:

- For Solaris: % `tar -xvf ThumbnailServer_Sol_<version>.tar`
- For AIX: % `tar -xvf ThumbnailServer_Aix_<version>.tar`
- For Linux: % `tar -xvf ThumbnailServer_Linux_<version>.tar`

This command creates or extracts the following four files.

- `thumbserver<platform>Setup.bin`
- `thumbserver_<platform>.zip`
- `ts_install.properties`
- `ts_config.properties`

6. Run the installer by entering `./thumbserver<platform>Setup.bin`.

The Thumbnail Server Installer splash screen is displayed, followed by the Welcome screen.

7. Click **Next**.

The license agreement terms are displayed.

8. Select the option to accept the terms of the license agreement, then click **Next**.
You are prompted as to whether you want the Thumbnail Server to run in HTTP mode or HTTPS (SSL) mode.
9. Select the appropriate option, then click **Next**.
Note: If you have chosen to run in HTTPS mode, then you must also configure Thumbnail Server for SSL. [Configuring Thumbnail Server for SSL](#), page 578 provides the instructions. The installer log file also contains this information.
10. Enter the number of a port that is available for Thumbnail Server or click **Next** to accept the default port (8081).
If you enter a port number already in use, you will be prompted to enter another port number.
11. Enter the number of a port that is available for Thumbnail Server administration or click **Next** to accept the default port (8008).
The installation summary screen appears, indicating the applications that will be installed.
12. Perform one of the steps:
 - If you are satisfied with the installation confirmation, click **Install**.
 - If you want to change a component of the installation, use the Back button to navigate back through the installer and change the install information.

The installation proceeds, installing the Thumbnail Server and its Configurator. The installer creates the following folders in the directory \$DM_HOME/thumbsrv/:

 - /conf
 - /container
 - /configurator
 - /install
 - /logs

The post-installation notification screen appears, informing you to run the Thumbnail Server Configurator to add support to serviceable repositories.
13. Make note of the Configurator location and click **Next**.
A message notifies you that Thumbnail Server was installed successfully.
14. Click **Done** to close the installer.

To configure repositories for Thumbnail Server on a non-Windows host:

1. Navigate to the Configurator folder identified in [Step 13](#) of the previous procedure (usually located in \$DM_HOME/thumbsrv).
2. Run the Configurator by entering `./thumbServer<platform>Configurator.bin`.
The Thumbnail Server Configurator splash screen is displayed, followed by the Welcome screen.
Note: Your Connection Broker and any repositories requiring Thumbnail Server must be running to use the Configurator.
3. Click **Next**.
The license agreement terms are displayed.

4. Select the option to accept the terms of the license agreement, then click **Next**.
 5. The installer searches for all repositories configured for your Content Server host (in the folder `$DOCUMENTUM/dba/config/`). Select a repository from the list and click **Next**.
You are prompted to enter the name of a repository SuperUser, their password, and the domain (optional). The Thumbnail Server will use this information to communicate with the repository.
 6. Enter the required information and click **Next**.
If the default AEK passphrase is used in the docbase configuration, then Thumbnail Server Configurator will automatically use the default AEK passphrase.
If the docbase is configured with the custom AEK passphrase, then the Thumbnail Server Configurator displays a screen for the user to enter the AEK passphrase.
 7. Enter the AEK passphrase.
If the repository you have chosen is part of a Distributed Content Server environment, a list of server config objects will appear for the repository.
 8. Select a server config object from the list and click **Next**.
A message notifies you that the repository was successfully configured for Thumbnail Server.
 9. Click **Done** to close the Configurator. The Thumbnail Server is started automatically.
Rerun the Configurator for any other repositories you wish to configure for Thumbnail Server.
- If you create a new repository on a host where Thumbnail Server is installed, you can run the Thumbnail Server Configurator to search for new repositories and configure them automatically.

Installing and configuring Thumbnail Server in silent mode

To install and configure Thumbnail Server in silent mode:

1. Extract the Thumbnail Server installer to a temporary folder.
2. Open the `ts_install.properties` file and verify if you want to change the default values in the file. This file is used for installing Thumbnail Server in silent mode.
3. Open the `ts_config.properties` file and update the values in the file. This file is used to run Thumbnail Server configuration in silent mode.
4. Navigate to the temporary folder, where the installer is extracted, in the shell prompt.
5. To start the silent installation, type:

- For Windows:

```
start /wait thumbserverSetup.exe -f c:\temp\ts_install.properties
```

Note: Use `start /wait` command to make the silent installation wait until the installation is complete.

- For Linux:

```
./thumbserverLinuxSetup.bin -f /temp/ts_install.properties
```

- For Solaris:

```
./thumbserverSolSetup.bin -f /temp/ts_install.properties
```

- For AIX:


```
./thumbserverAixSetup.bin -f /temp/ts_install.properties
```

Note: C:\temp in Windows and /temp in non-Windows mentioned in the preceding commands are the location of the properties files that are extracted from the installer.

6. Navigate to \$DM_HOME/thumbsrv/configurator folder in the shell prompt.

7. To run the Thumbnail Server configuration, type:

- For Windows:

```
start /wait thumbServerWinConfigurator.exe -f c:\temp\ts_config.properties
```

Note: Use *start /wait* command to make the silent installation wait until the installation is complete.

- For Linux:

```
./thumbServerLinuxConfigurator.bin -f /temp/ts_config.properties
```

- For Solaris:

```
./thumbServerSolConfigurator.bin -f /temp/ts_config.properties
```

- For AIX:

```
./thumbServerAixConfigurator.bin -f /temp/ts_config.properties
```

Note: C:\temp in Windows and /temp in non-Windows mentioned in the preceding commands are the location of the properties files that are extracted from the installer.

Installing and configuring Thumbnail Server on Docker environment

1. Install the supported version of Docker and Docker compose file in your host machine.
2. Set up the external database server and remote file system.
3. Installation and configuration of Thumbnail Server is bundled with Content Server image. So, configuration settings needs to be done along with Content Server configuration in the `statelesscs.conf` file. In the `statelesscs.conf` configuration file, by default, Thumbnail Server configuration is set to **NO** with default port numbers. Change the value to **YES** to enable the Thumbnail Server configuration. For example:

```
CONFIGURE_THUMBNAIL_SERVER = YES
THUMBNAIL_SERVER_PORT = 8081
THUMBNAIL_SERVER_SSL_PORT = 8443
```

4. Run the `stateless_config.sh` script.
5. To verify the configuration, perform the following:
 - Check the Thumbnail Server startup log at `$DOCUMENTUM/product/7.3/thumbsrv/container/logs/catalina.out` inside the container.
 - Check the Thumbnail Server URL (`http://:8081/thumbsrv/getThumbnail?`) availability from any browser.

By default, Thumbnail Server runs in HTTP mode on port 8081. The [Configuring Thumbnail Server for SSL, page 578](#) section contains more details if you want the Thumbnail Server configuration in SSL mode.

Configuring Thumbnail Server for SSL

To enable a Thumbnail Server for Secure Sockets Layer (SSL) encryption, you must configure it to run over HyperText Transfer Protocol over SSL (HTTPS).

The following procedure provides steps for a default SSL configuration. There are cases where a Certificate from the Certificate Authority may be required. Tomcat website provides information to configure Tomcat for SSL and to use the keytool.

To configure Thumbnail Server for SSL:

1. Create a certificate keystore by executing the following keytool command:

- for non-Windows:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

- for Windows:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA
```

Note:

- Use “changeit” as the password (or add the keypass attribute). You can set the keystore and keypass parameters to change the default file name and location or the password, if required. The keystore is stored in the “.keystore” file in the user’s home directory, by default, as determined by the user.home system property.
- It is recommended to use JDK 7 update 21 or earlier or JDK 8 to generate certificate keystore.

2. Open the server.xml file in a text editor. The file is located in the following directory:

- for non-Windows:

```
$DM_HOME/thumbsrv/container/conf
```

- for Windows:

```
%DM_HOME%\thumbsrv\container\conf
```

3. Locate the HTTPS connector and uncomment the <Connector> node. By default, the connector is on port 8443 and contains the attribute: SSLEnabled="true".

```
<Connector port="8443"
SSLEnabled="true"
keystoreFile="%keystore_location%"/>
```

4. To point the application server to the certificate keystore that contains the SSL certificate, add the following attribute:

```
keystoreFile="%keystore_location%"
```

Its value should point to the location where the keystore file is created in step 1 (for example: C:\Documents and Settings\myAdminUser\.keystore).

Note: If you used a different password than the default password (“changeit”) in step 1, add an additional attribute to the above <Connector> node:

```
keystorePass="{password}"
```

5. To avoid the listener conflict, comment the following line in the keystore file:

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on"/>
```

6. Uncomment the following line in the keystore file:

```
<Listener className=
"org.apache.catalina.security.SecurityListener"/>
```

7. Ensure that the non-HTTPS default connector port is different from the HTTPS connector port in the keystore file.

For example:

```
<Connector port="8088"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"/>
```

8. Enter the appropriate port value in the above section, if required (the connector port is 8443 by default).
9. Save and close the server.xml file.
10. *If you ran the Thumbnail Server in HTTPS mode during the Thumbnail Server installation, skip to the next step.* Otherwise, update the repository's THUMBNAIL_URL using the following DQL statement.

If the name of the thumbnail store is "thumbnail_store_01", then use the following query to update the thumbnail_url:

```
update dm_store object set base_url=
'https://<hostname>:<port>/thumbsrv/getThumbnail?'
where name='thumbnail_store_01'
```

Use the port number used in Step 8.

11. Restart the repository and the Thumbnail Server.
12. To validate the installation, load the URL specified in the query in Step 10 from your browser. On your first attempt to load the URL, you are prompted to accept the Certificate.

Unconfiguration of Thumbnail Server

Unconfiguration of Thumbnail Server manually

1. Run Thumbnail Server configurator.
2. Select **Remove Thumbnail Server support from a repository**. A list of configured repositories is displayed.
3. Select the repository you want to unconfigure.
4. Select **Next** and provide the system administrator username and password.
5. Select **Next**.

Unconfiguration of Thumbnail Server in silent mode

1. Open the ts_config.properties file and update the value for INSTALL_TYPE as REMOVE instead of ADD in the file.
2. Navigate to %DM_HOME%/thumbsrv/configurator folder in the shell prompt.
3. To run the Thumbnail Server configuration, type:
 - For Windows:

```
start /wait thumbServerWinConfigurator.exe -f c:\temp\ts_config.properties
```

Note: Use *start /wait* command to make the silent installation wait until the installation is complete.

- For Linux:

```
./thumbServerLinuxConfigurator.bin -f /temp/ts_config.properties
```

- For Solaris:

```
./thumbServerSolConfigurator.bin -f /temp/ts_config.properties
```

- For AIX:

```
./thumbServerAixConfigurator.bin -f /temp/ts_config.properties
```

Note: C:\temp in Windows and /temp in non-Windows mentioned in the preceding commands are the location of the properties files that are extracted from the installer.

Uninstalling Thumbnail Server from a Windows host

Use these instructions to uninstall Thumbnail Server from a Windows host.

1. Connect to the Content Server host as the Documentum Content Server installation owner.
2. Ensure that you have unconfigured all the repositories that were configured with Thumbnail Server.
3. Stop the Thumbnail Server.
4. Click **Start > Settings > Control Panel > Add/Remove Programs**.
5. In the Change or Remove Programs window, select **Documentum Thumbnail Server** and click **Change/Remove**.

The uninstaller for Documentum Thumbnail Server opens and displays the Welcome screen.

6. Click **Next**. The Summary screen displays the features that will be uninstalled.
7. Click **Uninstall**. The uninstallation proceeds.

A message informs you that Thumbnail Server was successfully uninstalled.

8. Click **Done**.

If you are going to reinstall the Thumbnail Server, you must restart the Thumbnail Server host before doing so.

Uninstalling Thumbnail Server from a non-Windows host

Use these instructions to uninstall Thumbnail Server from a non-Windows host.

To uninstall Thumbnail Server on non-Windows:

1. Connect to the Content Server host as the Documentum Content Server installation owner.
2. Ensure that you have unconfigured all the repositories that were configured with Thumbnail Server.

3. Stop the Thumbnail Server.
4. Navigate to the `$DOCUMENTUM_SHARED/uninstall/thumbserver` folder.
5. Run the `Uninstall` file by entering `./Uninstall`.
The Thumbnail Server Uninstaller splash screen is displayed, followed by the Welcome screen.
6. Click **Next**.
The uninstallation summary screen appears, indicating what will be uninstalled.
7. Click **Uninstall**.
The uninstaller removes any instances of the Thumbnail Server on the host, as well as the entire `$DM_HOME/thumbsrv` directory.
A message notifies you that Thumbnail Server was successfully uninstalled.
8. Click **Done** to close the uninstaller.

Stopping and starting Thumbnail Server on non-Windows

Use these instructions to stop and/or start Thumbnail Server on non-Windows.

To start or stop Thumbnail Server on non-Windows:

1. Open a command-line interface.
2. Navigate to `$DM_HOME/thumbsrv\container\bin\`.
3. To stop Thumbnail Server, type
`./shutdown.sh`
4. Navigate to `$DM_HOME/thumbsrv\container\bin\startup.bat`.
5. To start Thumbnail Server, type
`./startup.sh`

Verifying the Thumbnail Server installation

This section describes how to verify that you have successfully installed Thumbnail Server. The best way to verify installation is to import a test file to the repository using a Documentum client that uses Thumbnail Server. This assumes that at least one of Media Transformation Server, Audio-Video Transformation Server, or Document Transformation Server is installed, and is correctly generating thumbnail renditions. Consult the product documentation for those products.

Another verification method is to confirm that the Thumbnail Server is running.

To ensure your Thumbnail Server installation is working correctly, we recommend you complete both tests.

Verifying that Thumbnail Server is running

Once Thumbnail Server is started, you can verify that it is successfully serving default thumbnails through a web browser.

To verify that Thumbnail Server is running:

1. Open a web browser.
2. Enter the following address, based on your configuration:
`http://<hostname>:<port>/thumbsrv/getThumbnail?`

A default icon is displayed in the web browser.

Note: Ensure that the host name provided in the URL is accessible from the host where the browser is running.

Configuring Thumbnail Server in a trusted content store

To configure Thumbnail Server in a trusted content store:

1. Create a physical folder on the Content Server system.
2. Click **Administrator > Storage Management > Storage** and create a Location using Documentum Administrator and point to the physical folder created in the preceding step.
3. Create a file store with the following values:
 - Select the Location created in previous step.
 - Select **Yes** to encrypt the file store.
 - Select the media type as **Thumbnail Content** for the file store.
4. Run the following DQL queries in the doctbase:

```
update dm_filestore object set base_url='(select base_url from dm_filestore
where name = 'thumbnail_store_01')' where name = '<newly created file store>'

update dm_format object set default_storage = '<r_object_id of newly created
file store>' where name in ('jpeg_th','jpeg_lres','jpeg_story')
```
5. Restart the Content Server and Content Transformation Services server.
6. Verify if the thumbnails are stored in the newly created physical location of Content Server system and if the thumbnails are visible through any Documentum client.

Administration and Configuration

This section describes how to perform administration tasks using the Thumbnail Server.

Understanding default thumbnails

Documentum provides a set of default thumbnails, a default rules file, and the default_thumbnails.dtd. When you install the Thumbnail Server, the default rules file and DTD are stored in the following repository folder:

```
/System/ThumbnailServer
```

The default thumbnails are stored in this repository folder:

```
System/ThumbnailServer/thumbnails
```

The Thumbnail Server returns a default thumbnail rendition if it receives a URL from an application without a path or a store parameter. The server parses such URLs, and attempts to match the parameters it is given to a rule in the rules file. When a match is found, the associated default thumbnail is returned. If no match is found, the server returns a hard-coded default thumbnail. [Thumbnailing](#) provides more information on how thumbnails are requested and returned.

If the URL contains a storage location and path, the proper thumbnail is returned. Otherwise, the default thumbnail for this format type (based on its own mapping in the rules file) is returned. If it is unable to find the mapping, the default thumbnail is returned.

The rules in the rules file provided by Documentum reference four parameters:

- `object_type=type_name`
type_name identifies the object type of the returned object. This is the value in the object's `r_object_type` attribute.
- `format=format_name`
format_name identifies the format of the object's primary content object. This is the value of the format's `fdm_format` object.
- `is_vdm=boolean`
`is_vdm` indicates whether the object is a virtual document. The value is either true or false.
- `size=64`
 If `size` is specified as 64, then 64x64 default images will be served. Otherwise, the 32x32 default images will be returned.

For everything else that is not included in the rules file, the default thumbnail is returned.

You can modify the rules file to include rules that reference other parameters. [Adding default thumbnails](#), page 584 contains instructions for adding customized default thumbnails.

If an application uses the DQL THUMBNAIL_URL keyword to request a URL for an object that has no thumbnail rendition, Content Server returns the object type, format, and virtual document parameters for the object. The application then uses those parameters to build a URL that returns a default thumbnail.

The GET_FILE_URL administrative method does not return these parameters. If a thumbnail rendition does not exist, the method returns empty strings for the path and store argument. The application can send a URL without the path and store arguments, which causes the Thumbnail Server to return a default thumbnail. The application can also query the repository to obtain values for the parameters and build a URL using those values.

Adding default thumbnails

If you have custom object types, you may want to add default thumbnails for those types.

Observe the following guidelines when editing the `default_thumbnails.xml` file:

- Rules are processed in order until a match between the format and the thumbnail is found.
- The file must conform to the `default_thumbnails.dtd`.
- All file paths must be relative to:

`/System/ThumbnailServer/thumbnails`

- When a `<fixed_image>` or `<backup_image>` element includes the path to a file, be sure the file already exists.

When processing a URL, the Thumbnail Server returns an error if it does not find a file specified in a `<fixed_image>` or `<backup_image>` element.

- File paths specified in a `<pattern>` element are not required to refer to an existing file.

When processing a URL, the Thumbnail Server checks for the existence of a file specified in a `<pattern>` element. If the file is not found, the server returns the default thumbnail.

To add default thumbnails:

1. Import the thumbnail image to this repository folder:
`/System/ThumbnailServer/thumbnails` folders
2. Open the `default_thumbnails.xml` file (located at `/System/ThumbnailServer`) in a text editor.
3. Scroll down to the appropriate location in the file, as indicated by the preceding guidelines.
4. Following the guidelines indicated above, modify the `default_thumbnails.xml` file and add a rule for the new thumbnail.
5. Save and close the `default_thumbnails.xml` file.
6. Log in to Thumbnail Server.
7. Restart the Thumbnail Server service.

Changing the ticket time-out value

The time-out period for an encrypted ticket is set in the `web.xml` file. If required, you can change the value from the default of five minutes.

To change the ticket time-out value:

1. Log in to the Thumbnail Server host as an administrator.
2. Stop the Thumbnail Server service.
3. Navigate to the WEB-INF directory, located in:
`%DM_HOME%\thumbsrv\container\webapps\thumbsrv\WEB-INF`
4. Open the `web.xml` file in any text editor.

5. Scroll down to the `init-param` section:

```
<init-param>
  <param-name>ticket_timeout</param-name>
  <param-value>300</param-value>
</init-param>
```

6. Change the `<param-value>` element associated with the `ticket_timeout` parameter name. This will be the ticket time-out value, in seconds. The value must be numerical.
7. Save and close the `web.xml` file.
8. Restart the Thumbnail Server service.

Activating thumbnail logging

You can configure Thumbnail Server to log thumbnail requests. The logging feature will not be activated automatically upon installation of the Thumbnail Server -- it must be done manually. We strongly recommend that this feature be activated.

Logging may be useful for you for a number of reasons. It may help you troubleshoot any errors occurring with the Thumbnail Server. For example, if you are seeing some thumbnails but not others, examining the thumbnail server log may indicate a problem with the Thumbnail Server. The log file also indicates errors relating to invalid or expired tickets.

The Thumbnail Server log file, `localhost<yyyy-mm-dd>.log`, is located in this directory:

```
%DM_HOME%\thumbsrv\container\logs
```

To activate thumbnail logging:

1. Log in to the Thumbnail Server host as an administrator.
2. Stop the Thumbnail Server.
3. Navigate to this directory:

Windows

```
%DM_HOME%\thumbsrv\container\webapps\thumbsrv\WEB-INF
```

Non-Windows

```
$DM_HOME/thumbsrv/container/webapps/thumbsrv/WEB-INF
```

4. Open the `web.xml` file in any text editor.
5. Change the `param-value` for `param-name <debug>` to `TRUE`.

```
<param-name>debug</param-name>
<param-value>false</param-value>
```

6. Save and close the `web.xml` file.

7. Restart the Thumbnail Server service.

XML Store

This chapter is intended for system administrators who are responsible for deploying XML Store on Documentum Content Server.

To use this chapter, you need the following:

- Administrative privileges on the computer where you are deploying XML Store
- Working knowledge of Microsoft Windows or Linux
- Working knowledge of EMC Documentum Content Server administration and configuration, including high-availability (HA) configurations

For more information, see the following resources:

- *EMC Documentum Content Server Administration and Configuration Guide*
- *EMC Documentum xDB Administration Guide*

Introduction

XML Store is an optional module of Documentum Content Server that gives Content Server extended capabilities to store and process XML data in the repository by integrating it with Documentum xDB, a highly scalable, native XML database.

XML Store adds standards-based XQuery to the XML capabilities of Documentum Content Server. XML Store enables richer searches, reusing, and composing of content in a more flexible manner

XML Store offers fine-grained access to any content fragment without requiring that content be chunked or burst into individual content objects. This functionality means users throughout the enterprise can conduct richer searches, achieve more flexible reuse and content composition.

XML Store preserves XML content as is, without mapping XML to RDBMS table rows and columns. The XML structure is preserved, allowing users to efficiently and accurately query content at any level of detail (for example, individual elements, attributes, content objects, or metadata attributes), even on large information sets. As a native XML repository, XML Store provides performance advantages over relational databases and file systems through specialized XML indexing methods, caching, and architecture optimized for XML.

XML Store optimizes performance for XML content files and handles access to XML content using XQuery. The XML Store works with all other Content Server features, such as versioning, security, and lifecycles, including XML applications.

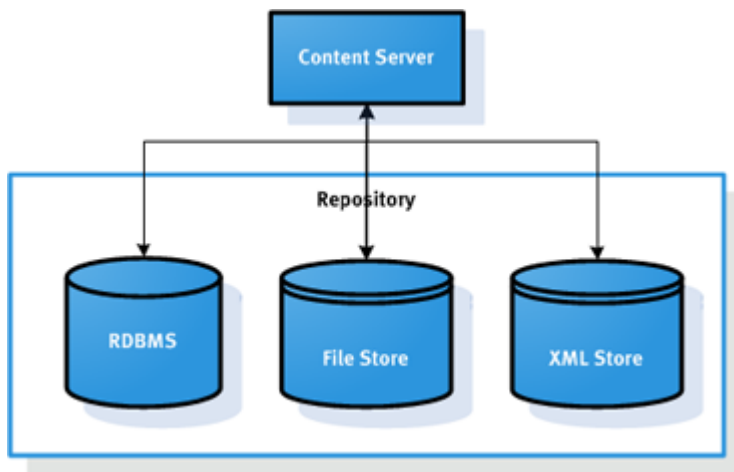
XML Store provides the following features:

- XML-specific storage type for storing XML content
- XQuery interface for searching and retrieving XML content and Documentum attributes through XQuery syntax.
- XML Store administrator interface to improve search performance by creating XML indexes and controlling the generation of segments and file/segment mappings.

XML Store enabled repository

XML Store is an optional module that adds a native XML database server to the Content Server repository. When you deploy XML Store for a Content Server, an xDB instance is deployed containing an xDB federation, which is logically a container of one or more xDB databases. A Content Server instance is associated with an xDB federation, and each xDB database in the federation maps to a repository in the Content Server. When you enable XML Store for a Content Server repository, a corresponding xDB database is created.

Figure 23. XML Store enabled repository



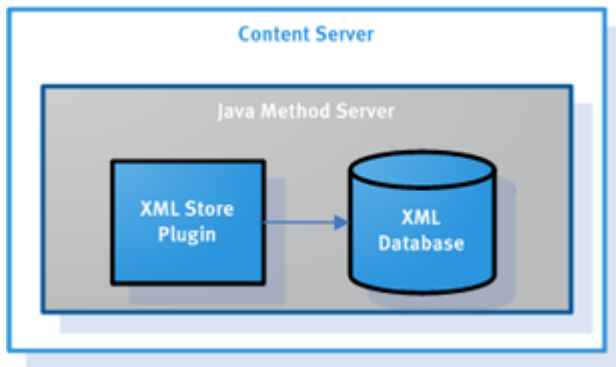
The *EMC Documentum xDB Administration Guide* provides the detailed information on Documentum xDB.

XML Store deployment modes

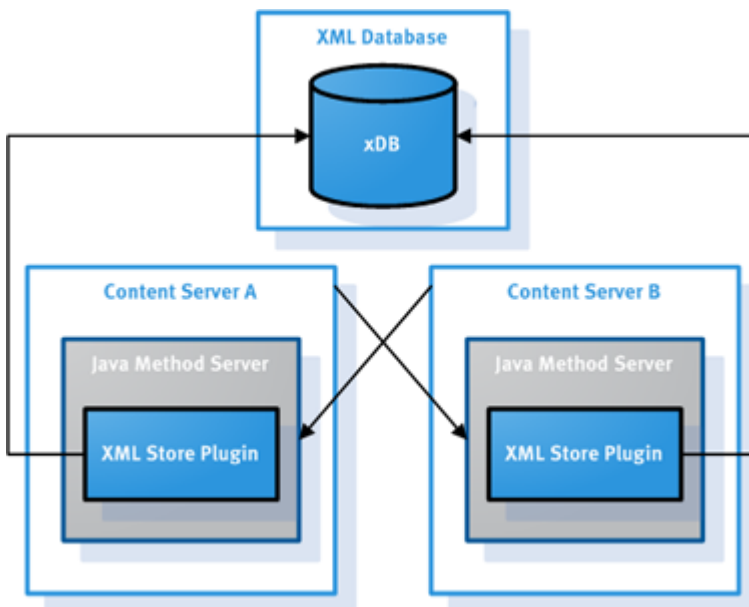
There are two ways you can deploy XML Store: use an embedded XML database managed by the Content Server or use an external XML database for high availability Content Server configuration.

Embedded XML database

Select this deployment mode if high availability is not required. It is called embedded because the XML database runs entirely within the Content Server's Java Method Server (JMS) and communicates over HTTP or HTTPS with the Content Server through an XML Store plug-in.

Figure 24. Embedded XML database**External XML database**

Select this deployment mode if you want to have the XML Store fail over to a secondary Content Server in the event the primary Content Server is unavailable. In this mode, you install a standalone xDB instance before deploying XML Store. provides the detailed information about high availability. XML Store uses Java Method Server failover mechanisms that allow methods to run on one or more servers when the Java Method Server for any one Content Server becomes unavailable. When a Java Method Server or Content Server is down, the XML Store can still be accessed and managed through other Content Servers. The configuration removes a single point of failure from any one Content Server node for XML Store, but does not provide failover for xDB itself. Additional steps are required to provide automatic failover for other aspects of the system. The *EMC Documentum xDB Administration Guide* provides the detailed information about configuring xDB for failover.

Figure 25. External XML database (xDB)

Pre-deployment tasks

Before you deploy XML Store, perform the following tasks:

- Obtain license keys for XML Store and Content Storage Services (CSS)

XML Store and Content Storage Services are both optional modules that require separate license keys. To store XML documents in an XML Store enabled repository, you must also enable Content Storage Services for the repository to configure assignment policies. You can do this before or during XML Store deployment.

- Install a standalone xDB instance (required for HA configurations)

If you plan to deploy XML Store in a high-availability (HA) environment using an external XML database, you must install a standalone xDB instance (version 10.2) first before deploying XML Store.

When you install xDB:

- Make note of the xDB host name, port number, superuser password, and data location during the installation. You will be required to provide the information when you deploy XML Store.
- During installation, choose **Advanced Settings** and select the **Other hosts may access the server** option.
- Ensure that the xDB client and xDB server have the xDB libraries with the same version.

The *EMC Documentum xDB Administration Guide* provides the detailed instructions on installing xDB.

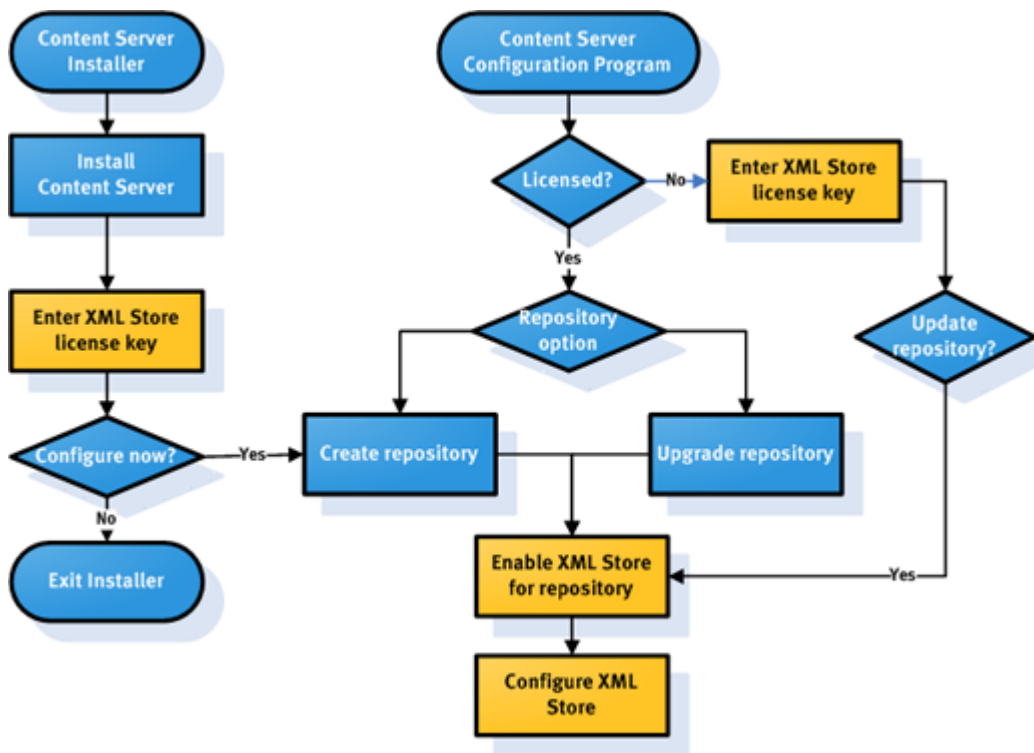
The installation program maintains an error log, which it writes to a file called `setupError.log` in the current working directory. If it cannot write into the working directory, it writes to the home directory of the user who initiated the installation. Reading this file may help you see what went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

Deploying XML Store

XML Store deployment workflow

XML Store is an optional module of Content Server that can be deployed during or after Content Server installation. After entering the XML Store license key, you can enable XML Store when you create a new repository or when you upgrade or update an existing one. The following diagram illustrates the general process for deploying XML Store for Content Server repositories. Steps specific to XML Store deployment are highlighted in yellow and will be explained in detail. provides the information about Content Server installation and configuration steps.

Figure 26. XML Store deployment workflow



Entering XML Store license key

As an optional module of Content Server, XML Store requires a separate license key. To enter the XML Store license key:

- Access the Content Server module licensing screen in one of these two ways:
 - During Content Server installation, Installer notifies you that Content Server has been successfully installed and asks you whether you want to launch the Content Server configuration program to configure a repository now. Select **Configure Now** and the Content Server configuration program runs and displays the Content Server module licensing screen.
 - After you install Content Server and exit Installer, launch the Content Server configuration program and select **Licensing**; then click **Next**. After you provide the installation owner password, the Content Server module licensing screen is displayed.
- In the Content Server module licensing screen, select **XML Store** and enter its license key.
If you have not activated the Content Storage Services module, also select **Content Storage Services** and enter its license key.
Click **Next**.
- The Content Server configuration program asks you whether you want to update any existing repositories with the new module. Click **Yes** to proceed to enable XML Store for a repository or **No** to exit.

Enabling XML Store for a repository

When the XML Store license key has been entered, you can enable XML Store for any repository in the Content Server.

Whether you are creating a new repository, upgrading a repository from an earlier version, or updating an existing repository, the Content Server Configuration Program will ask you whether to enable modules for the repository if they are not enabled yet.

To enable XML Store for a repository:

1. In the Content Server configuration program, when prompted to enable a module for the repository, select **XML Store**.
If the Content Storage Services module has not been enabled for the repository, also select **Content Storage Services**.
Click **Next**.
2. Specify a storage location for XML Store. The default location is `$DOCUMENTUM\data\xhive_storage`.
3. Select a deployment mode. provides detailed information about the two deployment modes.
 - **Embedded xDB instance managed by the Content Server:** Select this deployment mode when you deploy XML Store for just one Content Server.
 - **External xDB instance (required for high availability):** Select this deployment mode if you want to deploy XML Store to work with more than one Java Method Server or Content Server configured for failover in a high-availability environment. You must have already installed a standalone xDB instance.

If you select this option, provide the following xDB information:

- **xDB Host:** Enter the name or IP address of the xDB host.
- **xDB Port:** Enter the port number the xDB instance uses.
- **xDB Superuser Password:** Enter the xDB superuser password. This password is set when you install xDB and is used to create xDB databases in XML Store.
- **xDB Data Location:** Specify a valid path on the external xDB host where to store xDB data. This must be the same path specified for xDB during its installation.

If you select this option, you also need to perform additional configuration to deploy XML Store for multiple Content Servers. [Enabling XML Store to work with multiple Content Servers, page 593](#) provides more details.

4. If you select Embedded XML Database (xDB), provide the following information for XML Store.
 - a. Set the xDB superuser password. The superuser account will be used to create xDB databases in the XML store. Click **Next**.
 - b. Specify the following information for the XML Store:
 - **XML Store Port:** Enter an available port number to be used by the embedded xDB. The default is 1235.
 - **XML Store Directory Location:** Specify a directory location where the xDB database files will be stored. The default is `$DOCUMENTUM\data\xhive_storage`.

Click **Next**.

5. Enter the fully qualified domain name (FQDN) of the Content Server host.
6. The deployment process begins. When complete, exit the Content Server Configuration Program.

Enabling XML Store to work with multiple Content Servers

To deploy XML Store to work with multiple Content Servers, after you complete the deployment process for the first (primary) Content Server with an external XML database, you must perform some additional steps. provides information about XML Store deployment mode with an external xDB. If you deployed XML Store with an embedded XML database, you need to migrate data in the embedded XML database to an external XML database first and then perform additional steps. *EMC Documentum XML Store Administration Guide* provides more information.

To deploy XML Store to work with additional Content Server nodes:

1. Make sure that Content Server and Java Method Server instances have been properly configured for failover in a high availability environment.
Install additional Content Server nodes including the Content Server and WildFly components.
2. Copy the XML Store plugin XHiveConnector from the primary Java Method Server to additional Java Method Server.

- a. From the primary Content Server node, locate the deployed XhiveConnector.ear, located by default in \$DOCUMENTUM\jboss_directory\server\DctmServer_MethodServer\deploy.
- b. Copy the whole XhiveConnector.ear folder to the same location within the JBoss on additional Content Server nodes.
- c. If the JBoss is located in different locations on the additional Content Server node, verify the entries in the files dfc.properties and log4j.properties are correct for the additional Content Servers. The paths in these files must match the path for the additional Content Server nodes.

The files dfc.properties and log4j.properties are located by default in the following directory:

\$DOCUMENTUM\jboss_directory\server\DctmServer_MethodServer\deploy
\XhiveConnector.ear \APP-INF\classes

- Dfc.properties: The file has only one line:

```
#include E:\Documentum\config\dfc.properties.
```

You must make sure that this path is correct for the additional Content Servers.

- Log4j.properties: Evaluate all paths in this file to make sure that they are accurate for the additional Content Server nodes. This file's entries are similar to the following:

```
likelog4j.appender.F1.File=E:/Documentum/jboss_directory /server/DctmServer  
_MethodServer/logs/XhiveConnector.log
```

- d. Restart the Java Method Server.

3. On additional Content Servers, add XHiveConnector to the newly created `jms_config` object, using Documentum Administrator or by running a custom script.

- a. Launch Content Server Manager and start the IDQL utility; then use the following DQL query to obtain the server config ID, `r_object_id`:

```
1> ?,c,select r_object_id, object_name,  
server_config_id from dm_jms_config 2> go
```

The results returned are similar to the following:

```
r_object_id object_name 3d015b3880000102  
repo 3d015b3880000c57 xmlharcs_repo
```

Determine which server ID to use by examining the `object_name` that matches that of the `dm_jms_config` object on the additional node.

- b. Navigate to the `$DM_HOME\bin` directory and run the `dm_jms_admin` tool with the following parameters:

```
dm_jms_admin -docbase docbasename -username installowner -password password  
-action add -jms_host_name hostname -jms_port port -servlet_name XhiveConnector  
-base_uri /XhiveConnector/servlet/XhiveConnectorServlet -server_config_id serverconfigID  
-proximity 1
```

Determine which server ID to use by examining the `object_name` that matches that of the `dm_jms_config` object on the additional node.

- c. Navigate to the `$DM_HOME\bin` directory and run the `dm_jms_admin` tool with the following parameters:

```
dm_jms_admin -docbase docbasename -username installowner -password password  
-action add -jms_host_name hostname -jms_port port -servlet_name XhiveConnector  
-base_uri /XhiveConnector/servlet/XhiveConnectorServlet  
-server_config_id serverconfigID -proximity 1
```

For example:

```
dm_jms_admin -docbase repo -username Administrator -password Pass123  
-action add -jms_host_name secondaryCS -jms_port 9080 -servlet_name XhiveConnector  
-base_uri /XhiveConnector/servlet/XhiveConnectorServlet  
-server_config_id 3d015b3880000c57  
-proximity 1
```

4. Enable xHiveConnector on any additional Content Servers.

- The Content Server invokes a shared library or DLL to handle content stored in XML Store. The shared library or DLL is represented in the repository by a `dm_plugin` object. The shared library or DLL is stored as the content of the plugin object, and is created, by default, in `filestore_01` during Content Server installation.
- View information for this plugin object using the IAPI command: `API> retrieve,c,dm_plugin where object_name like '%xhive%' dump,c,1`. For XML Store to function properly on additional Content Servers in a distributed configuration, this plugin must be present in a distributed storage area accessible by all Content Servers.
- Use the DQL statement and information provided to ensure that each Content Server node has a properly configured file store component in a distributed storage environment. This step is required when storage is distributed. If `filestore_01` is accessible by all content servers, for example, through a UNC path, then a distributed store is not needed.
- [Implementing single-repository models, page 284](#) provides more information to complete the necessary steps. Specifically, refer to the step for moving the objects currently in `filestore_01` to the distributed store.

5. Verify the **Proximity** value for each **Target Host** on each Content Server. Because JMS failover does not support Remote Content Server configuration, the projection target range for each Content Server must be outside the range of 9000 to 9999. If the projection target falls in this range, JMS failover does not work as expected.

Check both the server.ini file and the server config object for each Content Server:

- a. Examine the server.ini file for each server, located by default in the Documentum directory, `.../dba/config/repositoryname/ server.ini` for the primary content server. Additional content servers .ini file is in the same directory as `server_machine_name_service_name.ini` For example:

```
[DOCBROKER_PROJECTION_TARGET] host = cshost1 port = 1489
#proximity=9001 [DOCBROKER_PROJECTION_TARGET_1] #host = #port = #proximity
= #host=cshost1 #port=1489 #proximity=9010
```

- b. Examine the server config object for each server in Documentum Administrator:
 - a. Select **Administration > Basic Configuration > Content Servers** , then right-click **Server configuration** and select **Properties**.
 - b. Select **Connection Brokers** and ensure the **Proximity** value for each **Target Host** falls outside the range of 9000 to 9999.