EMC® Documentum® xPlore

Version 1.1

Installation Guide

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

Table of Contents

List of Figures

List of Tables

Preface

This document provides instructions for installing, upgrading, and uninstalling Documentum xPlore.

Intended Audience

The reader knows the business nature of the indexing data and the network and operating system for xPlore.

Revision History

The following changes have been made to this document.

Revision Date	Description
April 2011	Initial publication for version 1.1

Additional documentation

This guide provides installation information. For information on planning, administration and development, refer to:

- *Documentum xPlore High Availability and Disaster Recovery Guide*
- *Documentum xPlore Administration Guide*
- *Documentum xPlore Development Guide*
- *Documentum xPlore Release Notes*

For additional information on Content Server installation and Documentum search client applications, refer to:

- *Content Server Installation Guide*
- *Documentum Search Development Guide*

Path conventions in this document

The following conventions are used in this guide:

- $DOCUMENTUM (Unix) and %DOCUMENTUM% (Windows): Environment variables that specify the Documentum Content Server root installation directory.

  Windows default: *C:\Documentum*. UNIX default: none.

- $DM_HOME (Unix) and %DM_HOME% (Windows): Environment variables that specify the Content Server installation directory and the product version subdirectory.

  Windows default: C:\Documentum\product*version_number*, where *version_number* is the version of Content Server. UNIX default: $DOCUMENTUM/*your_path*, where *your_path* is the path selected during installation, for example, *product/6.7*.

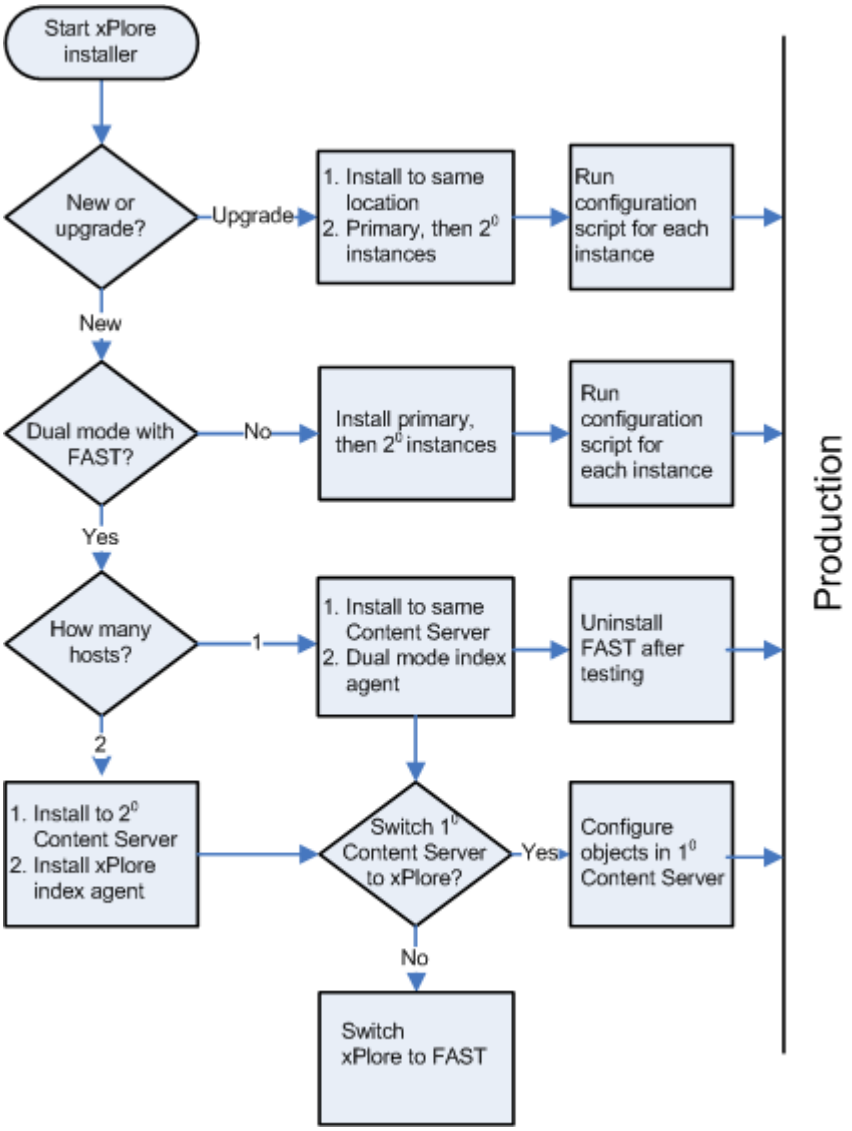
- *dsearch_home*: Installation directory for xPlore.

Chapter 1

Quick Start

If you are installing or upgrading xPlore in a test environment, Figure 1, page 9 gives an overview of your steps. For steps to migrate FAST to xPlore, see Chapter 7, Migrating from FAST to xPlore.

Figure 1. Installation and migration decision tree



Prerequisites —

- Update the Content Server with the latest hotfix for your environment.

Caution: If you do not install the hotfix, xPlore does not work properly.

- All instances must have access to storage areas.

1. Run the xPlore installer in a test environment. See Chapter 3, Installing xPlore.
2. Upgrade: Install xPlore to the same location as the previous version. Upgrade the primary instance first, with other instances stopped. Then upgrade secondary instances. Delete and reinstall/reconfigure index agents. See Chapter 5, Upgrading xPlore.
3. New install: Install and configure xPlore (Chapter 3, Installing xPlore) and an index agent (Chapter 4, Installing and Configuring an Index Agent).
4. Dual mode: Test xPlore and compare to FAST on one or two Content Servers. See Chapter 8, Testing xPlore and FAST in Dual Mode.

Requirements after installation —

- If you have installation problems, see the troubleshooting section (Chapter 9, Troubleshooting).
- Performance test your xPlore installation with the expected customer workloads, queries, and document profiles before moving your installation to production.

Chapter 2

Planning the xPlore Environment

For information on planning for and setting up backup and recovery, high availability, and disaster recover environments, refer to *Documentum xPlore High Availability and Disaster Recovery Guide*.

How xPlore works

Documentum xPlore (xPlore) is a multi-instance, scalable, high-performance, full-text index server that can be configured for high availability and disaster recovery.

Installation locations — The xPlore index service and search service are installed as a WAR file to a JBoss application server that is included in the xPlore installer. The administration console and online help are installed as war files in the same JBoss application server. The index is stored in the directory *dsearch_home*/data/repository_name/default/lucene-index or another configured location, such as a SAN or NAS.

xPlore instances — An xPlore instance is one installation of the xPlore WAR file in an application server container. The first instance that is installed is designated as the primary instance. You can have multiple instances on the same host (vertical scaling), although it is more common to have one xPlore instance per host (horizontal scaling). An instance can be configured to enable one or more of the following features:

- Content processing service (CPS)
- Indexing service
- Search service
- Administration console (includes analytics and instance and data management services)
- Spare

  A spare instance can be manually activated to take over for a disabled instance.

Domains — A domain is a separate, independent, logical grouping of collections within an xPlore installation. The Documentum index agent creates a domain and sends all indexed content to that domain in xPlore.

For more information on how xPlore processes documents and queries, see *Documentum xPlore Administration Guide*.

Planning your installation

1. Determine your sizing and performance requirements. See Determining requirements, page 12.
2. Determine your hardware requirements including xPlore server CPUs and memory, disk I/O, storage, network, and virtual machine requirements. See Hardware, page 13.
3. Determine your backup and recovery, high availability, and disaster recovery requirements.

   See *Documentum xPlore High Availability and Disaster Recovery Guide*.

Determining requirements

To size your full-text indexing system, use the xPlore sizing tool on Powerlink. Determine the following requirements:

- Indexing performance:
  - Indexing rate. The document indexing rate or document throughput is the rate at which new objects are added to the system or submitted for indexing. As the document complexity (for example, more words in a document or more words in a spreadsheet) increases, then the cost of creating the index increases. (More CPU is consumed and more disk I/Os are generated.) Higher throughput requirements result in higher processing costs.

    You can increase throughput by adding more instances and by increasing the speed and capacity of the machines feeding the data source to xPlore.
  - Indexing-to-query latency. Indexing-to-query latency is the time from when an object is saved to when the object is searchable. If you require a low indexing-to-query latency, an object must become searchable as fast as possible. Typically, this requirement is for environments in which many searches are performed and many objects are created and edited. In a fixed data scenario, large quantities of unchanging business data are stored but rarely or never modified, and a longer latency period is acceptable.

    A multi-instance installation provides a faster save-to-search time than a single instance.
- Query performance. Query performance is the speed at which results are returned for a query. Query performance depends on the following factors:
  - Number of users
  - Rate at which the users issue queries
  - Complexity of the queries, which impact disk I/O and CPU. A query is complex with wildcards, many terms in the query, and unselective terms.
  - Number of results

  - Permissions of the user (few permissions cause slower queries)
  - Size of the index
  - Hardware capacity
- Index size. The size of an index depends on the size of the largest documents indexed and the amount of indexable content in the documents. A large file can contain a small amount of indexable text and a large amount of unindexable content, such as graphics.

  As an index gets larger, the time it takes to merge the indexes grows because of the increase in disk I/O and CPU demand. Use a multi-instance installation for large documents, to spread the index over multiple hosts.

  To estimate your index size, you can index a subset of your most typical Content Server documents using an xPlore instance. Use the size of the sample index to estimate the size of the final index that would result from indexing all of your documents. For best results, use the xPlore sizing tool on Powerlink.

Sizing the xPlore system

EMC provides tools and guides for system sizing to help you determine the appropriate number of instances and system capabilities that you require. EMC Documentum Professional Services (or qualified third-party integrators) can assist you in reviewing your needs and completing the xPlore sizing spreadsheet. Indexing stops if the indexing host runs out of disk space or temp space.

Installations can range from simple to complex. For example, a simple installation has single-term searches and out-of-the-box latency for new documents that are medium-to-small size. A more complex installation has Boolean or wildcard searches, large result sets, low latency requirements, and large documents.

An installation on under powered machines or not appropriately configured can result in poor performance. Indexing fails if a disk fills up. To appropriately size your system, choose one of the following (the first is recommended):

- Use the xPlore sizing guide available on the download site and in Powerlink.
- Use the same hardware requirements that you established for your current FAST full-text indexing system.

Budget for staff to perform validation testing (including performance) on production data with real user scenarios. Always test xPlore in multi-user mode before installing it for production use.

Hardware

- Virtual machines, page 17
- Hardware tradeoffs, page 17

CPU

xPlore can have periods of CPU-intensive activity. To achieve higher indexing rates, some installations need additional CPUs for improved query response.

32-bit versus 64-bit environments

In some lower memory scenarios, 32-bit JVMs perform better than 64-bit JVMs, because the memory consumption for a single object is larger in 64-bit JVMs. Generally, 64-bit environments perform better than 32-bit.

In a 64-bit operating system, 4-6GB of memory for each xPlore instance can be easily reached. However, CPU usually reaches maximum usage before the system runs out of memory.

64-bit JVMs have these performance advantages over 32-bit JVMs:

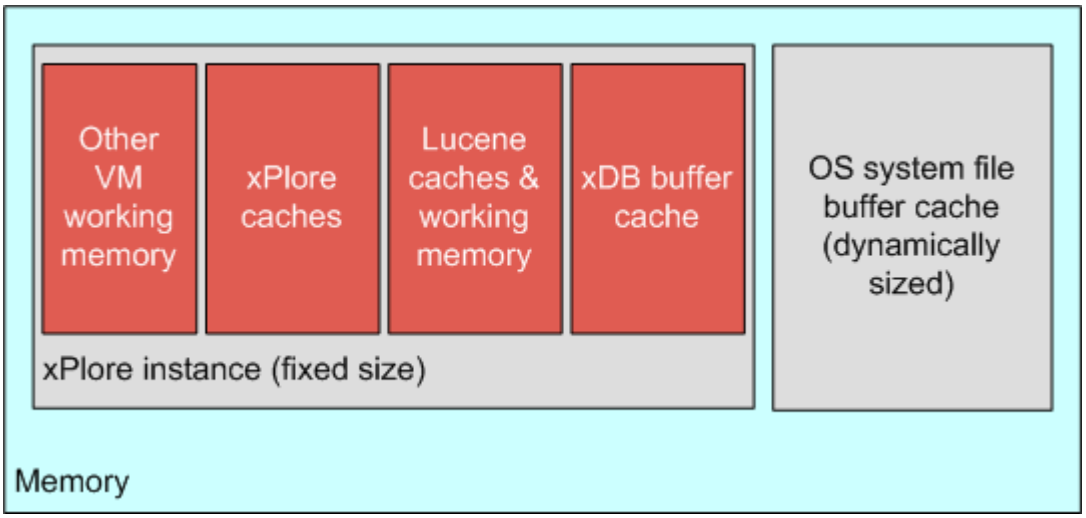
- More memory to cache index structures for faster query access
- Additional memory to index large documents
- Service higher query and indexing rates
- Each xPlore instance can use more than 2 GB of memory.

Memory

Full-text indexing is a memory-intensive operation. It is best to have the most memory you can afford on an xPlore instance. xPlore instance memory usage is tuned out-of-the-box. If you have large documents, millions of documents, or a high simultaneous query rate, you can tune your xPlore instance memory usage.

Figure 2, page 15 illustrates the relationship between xPlore instance and operating system memory usage. Total memory usage is divided between an xPlore instance and the operating system. Each xPlore instance has its own fixed size memory. This memory contains a Lucene cache and working memory, an xDB buffer cache, and optional virtual machine working memory. The operating system file buffer cache reserves memory space for temporary files, xDB data, and Lucene index structures. This cache is dynamic and expands to fill any remaining memory without interfering with xPlore processes.

Figure 2. xPlore memory usage



xPlore out-of-box parameters are adequate for most applications. For further memory tuning, Table 1, page 15 provides a description of xPlore memory pools and recommendations for tuning. Configuration of these caches is described in the *Documentum xPlore Administration Guide*, "System sizing and tuning" section.

Table 1. xPlore memory pools and tuning

Memory Pool	Description	Recommendations
xPlore Caches	For example, temporary result cache is used to buffer results to be sent to users.	None.
Lucene Caches and Working Memory	Used in processing queries. Uses application server JVM.	Higher query rates and larger numbers of documents require more memory.
xDB Buffer Cache	Used to store XML file blocks (ingest and query).	Increase its size for higher query rates.
Operating System File Buffer Cache	Temporary files, xDB data, and Lucene index structures. Although this cache will typically expand to fill any remaining memory, it will not interfere with xPlore processes.	Its largest impact is on Lucene (inverted index) performance.
Other VM Working Memory	Used to manage internal virtual machine operations	None.

Storage

- Multi-instance installations must store their data on the network in a Storage Area Network (SAN) or Network Attached Storage (NAS) device. At minimum, storage must be on a network

share. You can compensate for lower performing NAS by implementing a high performance network throughout your xPlore installation.

- When SAN remote mirrors are used, only one xPlore instance can run on the data (both indexing and querying at the same time). Two xPlore instances from two separate installations (xPlore federations), one local and one remote, cannot operate on the same index.
- Determine the type of disk system using these constraints and recommendations:
  — Keep disk I/O latency as low as possible.
  — If your indexing storage requirements are small, use direct-attached disks. For large indexes, use SAN devices and disk arrays for storing indexes and tokens.
  — Use a disk array in production environments, rather than a single disk, because a single disk creates a bottleneck for indexing.
- SANs typically represent virtualized storage. xPlore can be one of several applications attempting to leverage the virtual storage. As with virtual machines, ensure that the virtual storage is appropriately sized for all applications that access it.

Table 2. Storage areas

Files	Description	Indexing	Queries
xDB data	Stores DFTXML and other xDB data (metrics, auditing, ACLs, and groups)	For batch XML files, the next free disk block is consumed.	Random access retrieval for specific elements and summary
xDB redo logs*	Stores transaction information	Updates to the xDB data areas are logged first.	When snapshot information is required
Lucene indexes	Query lookup and retrieval of facets and security	Index information updated through inserts and merges	Inverted index lookups and facet and security retrieval
Lucene temp area	Stores updated Lucene indexes (non-transactional data)	Uncommitted data stored to log	N/A
index agent content staging area	Temporarily stores content during indexing process	Temporarily stores content during indexing process	N/A

* Old xDB redo logs are purged at each backup. For incremental backups, set the `keep-logfiles` parameter to true and allocate more disk space.

Network

Follow these guidelines when implementing your xPlore network connections:

- When using NAS, ensure that your network bandwidth is high and use jumbo frame support.
- Use Gigabit Ethernet for iSCSI in production environments. Do not use anything less than 1 Gb (for example, 100BaseT); 10 Gb Ethernet is good.
- Use high-capacity SAN Fibre Channel interconnects to the disk arrays.

Virtual machines

Because many applications are not busy all the time, you can install many of them on the same machine using virtual machines. In addition, virtual machines offer easier installation and recovery options than native installations. Potential drawbacks with virtual machines:

- xPlore is very CPU- and disk I/O-intensive, and it can consume more resources than the average virtual machine.
- Other applications on a virtual machine host compete with xPlore and can cause xPlore performance to fluctuate.
- The performance of virtual CPUs is not as good as the performance of native CPUs for xPlore CPU-intensive workloads.
- The maximum number of CPUs supported by a virtual machine product is lower than a hardware environment. For example, VMWare ESX 3.5 supports up to eight cores. If you have fewer CPUs per operating system require more xPlore instances to perform the same work as in a native environment.

Hardware tradeoffs

Table 3, page 17 illustrates how your environment and xPlore configuration options affect index size, indexing performance, and query performance. The table suggests which resources you can increase or improve to alleviate the reduction in service.

Table 3. Hardware budget: Performance and size

		Significant Impact:			
	Environment or configuration	**Index storage**	**Indexing performance**	**Query performance**	**Resources consumed**
Indexable documents	Document size	+	+	x	Disk space, memory

		Significant Impact:			
	Environment or configuration	**Index storage**	**Indexing performance**	**Query performance**	**Resources consumed**
	Document complexity	+	+	x	Disk space, CPU, memory, network bandwidth
	Number of ACLs and groups	+	+	+	Disk space, memory
	Query complexity	N/A	x	+	CPU, memory, Disk I/O
	Total number of documents	+	+	+	Disk space, disk I/O, CPU.
Ingestion rate		N/A	+	N/A	Disk I/O, CPU
Query rate		N/A	N/A	+	Disk I/O, memory, CPU, more instances
xPlore configuration	Number of collections	N/A	+	+	
	Lemmatization	+	+	+	Disk space
	Diacritics	+	+	+	

- +: Has a significant impact on index size, indexing performance, or query performance.
- *x*: Does not have a significant impact.

Chapter 3

Installing xPlore

You can perform the following tasks to install or upgrade xPlore:

If you are upgrading xPlore, see Chapter 5, Upgrading xPlore. If you have a previous installation of the FAST indexing server, see Chapter 7, Migrating from FAST to xPlore.

After you install xPlore, install and configure an index agent. See Chapter 4, Installing and Configuring an Index Agent

Fulfilling requirements

Installation owners — Installation owners of Content Server and xPlore (including CPS) must be identical.

UNIX/Linux platforms — Installation of xPlore in console mode is not supported. Set up your environment to support GUI installation.

Enabling SSL

Use the following procedure to enable SSL connections to xPlore administrator and between xPlore and the Content Server.

1. Enable an HTTPS port for the xPlore instance in JBoss. Refer to JBoss documentation.

2. Create a certificate keystore using the Java keytool utility:
   - For UNIX:

     ```
     $JAVA_HOME /bin/keytool -genkey -alias tomcat -keyalg RSA -keystore my.keystore
     ```

   - For Windows:

     ```
     %JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA -keystore my.keystore
     ```

   For the CN value (first and last name), enter in the hostname of the xPlore instance. This must be the same hostname you used to configure the index agent connection to xPlore. Use *changeit* as the password and *tomcat* as the alias. The keystore is stored in the user home directory by default (user.home system property).
3. Copy your keystore file to *dsearch_home*/jboss4.3.0/server/DctmServer_PrimaryDsearch/conf.
4. Install the certificate file in the Java runtime security lib. Use the following command:

   ```
   JAVA_HOME\bin\keytool -importkeystore –srckeystore
   keystorePath -destkeystore jreCACERTSpath
   ```

   Replace *keystorePath* with the path to my.keystore that was generated in step 2. Replace *jreCACERTSpath* is *dsearch_home*/jboss4.3.0/jdk/jre/lib/security/cacerts.
5. Instruct JBoss to use the certificate and use the secure port. Modify the file *dsearch_home*/jboss4.3.0/server/DctmServer_PrimaryDsearch/deploy/jboss-web.deployer/server.xml and uncomment the Connector element for port 9302. Set SSLEnabled to true and keystoreFile to the full path on the host:

   ```
   <Connector port="9302" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    keystoreFile=
    "dsearch_home/jboss4.3.0/jdk/jre/lib/security/cacerts"
    keystorePass="changeit"
    clientAuth="false" sslProtocol="TLS" />
   ```

6. (Optional) To disable HTTP, comment out the following Connector entry in the file *dsearch_home*/jboss4.3.0/server/DctmServer_PrimaryDsearch/deploy/jboss-web.deployer/server.xml:

   ```
   <Connector port="9300" address="${jboss.bind.address}"
    maxThreads="250" maxHttpHeaderSize="8192"
    emptySessionPath="true" protocol="HTTP/1.1"
    enableLookups="false" redirectPort="9302" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true" socketBuffer="64000"/>
   ```

7. Change the dm_ftengine_config object in the Content Server to set HTTPS as the value of dsearch_qrserver_protocol.
   a. Get the dm_ftengine_config object:

      ```
      retrieve,c,dm_ftengine_config
      dump,c,l
      ```

   b. Find the index value of dsearch_qrserver_protocol (for example, 6) and the index value of dsearch_qrserver_port (for example, 3).
   c. Set the following values. This example uses the index values returned by the dump command:

      ```
      set,c,param_value[6]
      HTTPS
      set,c,param_value[3]
      9302
      ```

8. Edit indexserverconfig.xml in *dsearch_home*/config. Find the node element and change the value of the url attribute to match the HTTPS-enabled port. The hostname must be the same hostname you used to configure the index agent connection to xPlore. For example:

   ```
   <node appserver-instance-name="PrimaryDsearch" xdb-listener-port="9330"
   primaryNode="true" status="normal" url="https://localhost:9302/dsearch/"
   admin-rmi-port="9331" hostname="PLEENGBATCHM" name="PrimaryDsearch">
   ```

   Note: On Windows 2008, you cannot save the file with the same name, and the extension is not shown. By default, when you save the file, it is given a .txt extension. Be sure to replace indexserverconfig.xml with a file of the same name and extension.

9. Scripting support: Change the port entry to match the new HTTPS-enabled port in the file xplore.properties, located in located in *dsearch_home*/dsearch/admin. Change the protocol entry from *http* to *https*.

10. Edit the JBoss service config xml file that is located at *dsearch_home*/jboss4.3.0/server/serviceConfig/instance_name/conf, for example:

    ```
    C:\xPlore\jboss4.3.0\server\serviceConfig\PrimaryDsearch\conf\DmPrimaryDsearch.xml
    ```

    Change the port in the heartbeat url from the HTTP port number to the HTTPS port number. In this example, it is changed from 9300 to 9302:

    ```
    <heartbeat>
    <url>https://server-vm7:9302</url>
    </heartbeat>
    ```

11. Restart the Content Server, method server, and all xPlore instances.

12. Validate SSL by navigating to the JBoss xPlore test page in your browser. The *hostname* must be the same hostname you used to configure the index agent connection to xPlore.

    ```
    https://hostname:9302/dsearch
    ```

    Troubleshooting: An HttpHostConnectionException indicates that one of the steps was not configured properly.

Installing xPlore with Content Server 6.7

Documentum Content Server version 6.7 has built-in support for xPlore. Follow these instructions. To enable the FAST indexing server, see Enabling FAST with Content Server 6.7, page 46.

1. See Fulfilling requirements, page 19.
2. Deploy or upgrade to Content Server 6.7. Configure at least one Documentum repository and one connection broker. See Chapter 5, Upgrading xPlore.
3. Install the latest cumulative hotfix for your version of the Documentum Content Server.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: On AIX and Solaris, use GNU tar to untar the cumulative hotfix; otherwise, in the Jboss directory, the file names are truncated.

4. Install the xPlore primary instance. See Installing the primary instance, page 22.
5. (Optional) Install secondary instances. See Installing a secondary or spare instance, page 24.

6. For each Documentum repository, install and configure a Documentum index agent for xPlore. See Chapter 4, Installing and Configuring an Index Agent.

Installing xPlore with Content Server 6.5 SPX or 6.6

1. See Fulfilling requirements, page 19.
2. Install the latest cumulative hotfix (patch) for your version of the Documentum Content Server. This hotfix installs support for xPlore.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: On AIX and Solaris, use GNU tar to untar the cumulative hotfix; otherwise, in the Jboss directory, the file names are truncated.
3. Content Server 6.5 SPX or 6.6 only:
   a. Run the script fulltext_setup_for_dss.ebs to create objects for xPlore. The script is located in `%DM_HOME%\install\admin` (Windows) or `$DM_HOME/install/admin` (Unix) on the Content Server machine.

      ```
      dmbasic -f fulltext_setup_for_dss.ebs -eSetupForDSS -- repositoryName
      username password fulltextHome
      ```

      - *repositoryName*: Name of the repository to set up for xPlore.
      - *username* and *password*: Username and password for repository installation owner.
      - *fulltextHome*: Fulltext directory on the Content Server, for example, `/export/user1/Documentum/fulltext` (Linux) or `C:\Documentum\fulltext` (Windows).

      Note: Enter a space before and after the double hyphen. Do not use environment variables in paths. For example:

      ```
      dmbasic -f fulltext_setup_for_dss.ebs -eSetupForDSS -- testenv
      tuser1 mypwd C:\Documentum\fulltext
      ```
4. Restart the Content Server.
5. Install the xPlore primary instance. See Installing the primary instance, page 22.
6. (Optional) Install secondary instances. See Installing a secondary or spare instance, page 24.
7. For each Documentum repository, install and configure a Documentum index agent for xPlore. See Chapter 4, Installing and Configuring an Index Agent.

Installing the primary instance

Prerequisites — For all xPlore instances, index agents and CPS instances:

- Install a primary instance before installing secondary instances. Secondary instances are optional.
- The same installation owner is used to install them, to avoid permission-related errors.

- They are all installed on the same operating system (Windows or Linux).
- On Windows machines, use universal naming convention (UNC) to specify shared folders (do not use mapped drives).

Note: On 64-bit Windows Server 2008, run these executables on a command line and click **Allow** when prompted.

1. Unzip one of these files to a temporary directory:

   Documentum_xPlore_1.1_linux_x64.zip (64-bit)
   Documentum_xPlore_1.1_linux.zip (32-bit)
   Documentum_xPlore_1.1_windows_x64.zip (64-bit)
   Documentum_xPlore_1.1_windows.zip (32-bit)

2. Run setup.exe (Windows) or setup.bin (Linux).
   1. Where to install xPlore: Choose a path that does not contain spaces. xPlore and its components are installed to a root directory. Default: C:\xPlore on Windows or $HOME/dss on Linux.
   2. On the **Get Watchdog Administrator Information** page, enter an SMTP email server and email address. The fields can be left blank. The email server cannot be a Microsoft Exchange server. You can set the SMTP server and email address at a later time in the file dsearch-watchdog-config.xml. This file is located in *dsearch_home*/watchdog/config. Set the properties of the *SendMailTask* task.

      The xPlore watchdog service is a Windows service or daemon process (a standalone Java process) that is installed on each xPlore host. The watchdog service monitors and checks the status of various xPlore processes and sends an email notification to the administrator.
3. Run *dsearch_home*\setup\dsearch\configDsearch.bat (Windows) or *dsearch_home*/setup/dsearch/configDsearch.sh (Linux).
   a. **Select Configuration Mode**: Choose **Create Primary Server Instance**.
   b. (Windows only) **Get Installation Owner Password**: **Installation Owner Password** is required for setting up the Windows service for this instance. **Installation Owner Domain** is the domain for the repository with which you want to associate this xPlore instance. Default: The current host name.
   c. **Server Instance Information**: **Server Name** must be unique in your xPlore federation of instances (default *PrimaryDsearch*). **Base Port** default is *9300*. The next 100 consecutive ports must be available. **Password for Admin User** is also the xDB Administrator password.
   d. **Dsearch Data and Config Directory Information** directories. For a single-instance environment, specify a local directory on this host for better performance. For multiple instances, these directories must be accessible and writeable by all xPlore instances.

      Data directory stores the primary instance xDB transaction log and the default collection storage area. (The storage area is different from the index agent temporary storage location.) Default: *dsearch_home*/data.

      Configuration Directory stores configuration information. Default: *dsearch_home*/config.
4. To start the primary xPlore instance, run the script in *dsearch_home*/jboss4.3.0/server/: start*PrimaryDsearch*.cmd or .sh. On Windows, you can also start the service **Documentum xPlore PrimaryDsearch**.

5. To test whether the primary instance is running, open your web browser and enter the following URL. *host* is the DNS name of the primary instance host and *port* is the xPlore port (default 9300).

   `http://host:port/dsearch`

   If the instance is running, you see a message like the following:

   ```
   The DSS instance PrimaryDsearch [version=1.1.0000.0159] normal
   ```

Note: If you install the 64-bit version of xPlore, both 32- and 64-bit JDK versions are installed. You can switch to 32-bit version later if desired. The root directory for the 32-bit version of the JDK is *dsearch_home*/jboss4.3.0/jdk.32.

Installing a secondary or spare instance

Prerequisites — For all xPlore instances, index agents and CPS instances:

- Install a primary instance before installing secondary instances. Secondary instances are optional.
- The same installation owner is used to install them, to avoid permission-related errors.
- They are all installed on the same operating system (Windows or Linux).
- On Windows machines, use universal naming convention (UNC) to specify shared folders (do not use mapped drives).

Note: On 64-bit Windows Server 2008, run these executables on a command line and click **Allow** when prompted.

1. Unzip one of these files to a temporary directory:

   Documentum_xPlore_1.1_linux_x64.zip (64-bit)
   Documentum_xPlore_1.1_linux.zip (32-bit)
   Documentum_xPlore_1.1_windows_x64.zip (64-bit)
   Documentum_xPlore_1.1_windows.zip (32-bit)

2. Run setup.exe (Windows) or setup.bin (Linux).
   - Where to install xPlore: Choose a path that does not contain spaces. xPlore and its components are installed to a root directory. Default: C:\xPlore on Windows or $HOME/dss on Linux.
   - On the **Get Watchdog Administrator Information** page, enter an SMTP email server and email address. The fields cannot be blank. They do not have to reference a valid server or email address. The email server cannot be a Microsoft Exchange server.

     The xPlore watchdog service is a Windows service or daemon process (a standalone Java process) that is installed on each xPlore host. The watchdog service monitors and checks the status of various xPlore processes and sends an email notification to the administrator.

3. Start the primary instance:
   - Windows: The **Documentum Search Services *PrimaryDsearch* (Primary)** Windows service, or *dsearch_home*\jboss4.3.0\server\startPrimaryDsearch.cmd
   - Linux: *dsearch_home*/jboss4.3.0/server/startPrimaryDsearch.sh
4. Run *dsearch_home*\setup\dsearch\configDsearch.bat (Windows) or *dsearch_home*/setup/dsearch/configDsearch.sh (Linux).

   a. **Select Configuration Mode**: Choose **Create Secondary Server Instance**.

If you are installing a secondary instance for backup, check **Create as a Spare Node**. To ensure that a spare instance can automatically substitute for a failed instance, all instances must share all data store paths.

Caution: If the spare is a backup for the primary instance, check **Dsearch Administration Console** on the **Server Instance Information** page.

b. (Windows only) **Get Installation Owner Password**: **Installation Owner Password** is required for setting up the Windows service for this instance. **Installation Owner Domain** is the domain for the repository with which you want to associate this xPlore instance. Default: The current host name.

c. **Server Instance Information**: **Server Name** must be unique in your xPlore federation of instances (default *DsearchNode2*). **Base Port** default is *9300*. Change this port for a secondary instance. The next 100 consecutive ports must be available. For **Password for Admin User**, use the xPlore administrator password.

d. **Primary Host Information**: **Primary Host** is the DNS name of the primary instance host, and **Primary Port** default is `9300`.

e. **Dsearch Log Directory (Secondary Node) Transaction Log Directory**: Specify a directory for transactional log files (default: *dsearch_home*/dblog). If you do not require sharing or high availability in a multi-instance configuration, specify a local directory on this machine for better performance. Make this directory accessible and writeable from the primary instance. On Windows, specify a UNC-enabled (Universal Naming Convention) path.

5. Restart all instances including the primary instance. To start a secondary instance, run the script in *dsearch_home*/jboss4.3.0/server/: start*Secondary_instance*.cmd or .sh. On Windows, you can also start the service **Documentum Search Services Secondary_instance**.

   Note: Documents are not indexed into the secondary instance until you create a collection and bind it to the secondary instance. Use xPlore Administrator to create collections.

6. Test whether the secondary instance is running: Open your web browser and enter **`http://host:port/dsearch`** using the host and port information for your secondary instance.

   If the instance is running, you see a message like the following (*DsearchNode2* is the default instance name):

   ```
   The DSS instance DsearchNode2 [version=1.1.0000.0159] normal
   ```

Installing Content Processing Service (CPS) instances

Prerequisites — For all xPlore instances, index agents and CPS instances:

- Install a CPS instance on the same operating system as other xPlore instances.
- Install a primary instance before installing a secondary CPS instance.
- The same installation owner is used to install them, to avoid permission-related errors.

- They are all installed on the same operating system (Windows or Linux).
- On Windows machines, use universal naming convention (UNC) to specify shared folders (do not use mapped drives).

By default, a CPS instance is installed when you install a primary or secondary instance. You can add CPS instances to improve performance and scalability.

1. If you are installing a CPS instance on a machine on which you have not installed xPlore, run the installer. See steps 1 and 2 of Installing a secondary or spare instance, page 24.
2. Run *dsearch_home*\setup\dsearch\configDsearch.bat (Windows) or *dsearch_home*/setup/dsearch/configDsearch.sh (Linux).

   Note: On 64-bit Windows Server 2008, run these executables on a command line and click **Allow** when prompted.
3. **Select Configuration Mode**: Choose **Create Content Processing Service only**.
4. **Server Instance Information**: **Server Name** must be unique in your xPlore federation of instances (default *CPS*). **Base Port** default is *9300*. Change this default for a secondary CPS instance. The next 100 consecutive ports must be available. Use the same password as the xPlore administrator.
5. On the **Primary Host Information** page, **Primary Host** is the DNS name of the primary instance host, and **Primary Port** default is 9300. You do not see this page if you are on the same host as the primary instance.
6. Verify that the export_path location in the remote CPS configuration.xml file is accessible by all instances. This file is located in the CPS host directory *dsearch_home*/dsearch/cps/cps_daemon. If you change this file, restart all instances including the remote CPS.

Starting the CPS instance

1.
   - Windows:
     — The **Documentum Search Services *CPS_instance* (CPS)** Windows service, or
     — *dsearch_home*\jboss4.3.0\server\start*CPS_instance*.cmd
   - Linux: *dsearch_home*\jboss4.3.0\server\start*CPS_instance*.sh

   where *CPS_instance* is the value you specified for **Server Name** you specified for the CPS instance.
2. Register the remote CPS instance in xPlore administrator. Open **Services > Content Processing Service** in the tree and then click **Add**. Enter the URL to the remote instance using the following syntax:

   http://*hostname:port*/services
3. To test the remote CPS service using the WSDL testing page, open a browser and specify the following syntax:

   http://*hostname:port*/services/cps/ContentProcessingService?wsdl

   A page showing the CPS service configuration is displayed.

Deleting an xPlore or CPS instance

Caution: Do not delete the primary instance before you have deleted all secondary instances.

1. The primary instance and the instance to be deleted must be running.
2. On the machine on which the instance is installed, run *dsearch_home*`\setup\dsearch\configdsearch.bat` (Windows) or *dsearch_home*`/setup/dsearch/configdsearch.sh` (Linux) and follow the instructions.

   Note: *dsearch_home* is the xPlore root directory.
3. On the **Select Configuration Mode** page, select the **Delete Existing Server Instance** option.
4. On the **Delete Instance** page, in the **Instance** field, select the name of the instance to delete.
5. On the **Select instance binding for internal collections** page, choose how to manage the instance internal system data:
   - **Delete** — (Default) Deletes all of the instance internal collections.
   - **Change Binding to Other Instance** — Select another instance to which to bind all of the instance internal collections. That other instance must be running. Make sure that you do not bind too many internal collections to a single instance.
   - **Ignore (you have already cleaned it up)** — No action is taken. Choose this option only if you have already deleted or bound this instance internal collections to another instance; otherwise, an error occurs.

Chapter 4

Installing and Configuring an Index Agent

Run the xPlore installer to install an index agent. Then configure the agent as described in Configuring an xPlore index agent, page 29.

For information on using the index agent UI after installation, refer to *Documentum xPlore Administration Guide*.

Configuring an xPlore index agent

Install and configure an index agent for each repository.

Prerequisites —

- The same installation owner is used to install all xPlore instances, index agents, and CPS instances, to avoid permission-related errors.
- All instances are all installed on the same operating system (Windows or Linux).
- Install xPlore if it is not already installed on the index agent host. See Installing the primary instance, page 22.
- Verify that the xPlore primary instance, Documentum Content Server, and Documentum global registry are started. After successfully starting up Content Server and the connection broker, wait several minutes (typically 15 minutes) before configuring the Documentum index agent.

You can configure multiple index agents.

1. Run *dsearch_home*`\setup\indexagent\configIndexagent.bat` (Windows) or *dsearch_home*`/setup/indexagent/configIndexagent.sh` (Linux).
2. Upgrade only: Delete the existing index agent and restart the index agent configuration.
3. (Windows only) **Get Installation Owner Password**, specify the password and domain (default: current host name).

4. **Server Instance Information**
   - **Server Name**: Unique name for this index agent.
   - **Base Port**: Port number (by default, 9200) for HTTP requests. The installer validates that the next 20 consecutive ports are available. Index agents on the same machine must have unique port numbers.
   - **Password for Admin User**: You can use the same password as the xPlore administrator.
5. **Primary Dsearch Server Information**: Specify the DNS name and port (default: *9300*) of the xPlore primary instance.
6. **Enter the directory for local content area**: Directory path to a directory in which content is temporarily stored during indexing. The installation owner must have write permissions on this directory.
7. **Connection Broker Information**: Click **Connect** after you supply the DNS name and port for the Documentum connection broker.
8. If the index agent configurator successfully connects to the connection broker, you see additional fields. Enter this information:
   - **Select Docbase**: Choose a Documentum repository to which the index agent connects.
   - **User Name**: Specify a valid superuser account for the repository.
   - **Password**: Specify the password for the repository superuser account.
9. **Global Registry Information**: Supply login information and global repository name.
10. **Select Storage Location**: Select the location to store indexes. (The storage location for indexes is different from the temporary content location that you previously specified for the index agent.) The same storage location is used for all index agents on the same machine.

    Use the xPlore Administrator to create storage locations. See the *xPlore Administration Guide*.
11. If you are installing the index agent in a multi-instance xPlore environment, restart all xPlore instances.

Starting the index agent

1. Windows: Start the index agent Windows service (or restart it, if it is already running) **Documentum *Index_agent***, or run *dsearch_home*\jboss4.3.0\server\start*Index_agent*.cmd.

   Linux: Run *dsearch_home*/jboss4.3.0/server/start*Index_agent*.sh (Linux).

   Index_agent is the value you specified for **Server Name**.
2. Set up index agent filters before you start reindexing. Otherwise, the filters cannot be applied to objects that already been indexed. You can install index agent filters that exclude cabinets, folders, or object types from being indexed. See "Using the index agent filters" in *xPlore Administration Guide*.
3. Start the index agent UI. In a browser, enter a URL that contains the index agent host DNS name and port (default *9200*):

   `http://host:port/IndexAgent/login_dss.jsp`

4. Log in as a valid user in the repository. The optional domain name is the name of an xPlore domain. If blank, the index agent creates a domain with the name of the repository.
5. If the index agent is stopped, choose one of these modes:
   - **Start Index Agent in Normal Mode**: Normal mode indexes content that is added to or modified in the repository from this point onwards.
   - **Start new reindexing operation**: (Migration mode) Starts the index agent and indexes all content in the repository (also called *refeeding* or *recrawling*).

     Object types that are registered for full-text indexing in dmi_registry are indexed, then dm_acl and dm_group objects are indexed.
   - **Continue: Reindex date time**: Continue indexing. *date* and *time* indicate when you stopped indexing.
6. If the index agent is running, choose one of these actions:
   - **Details**: Statistics and status of indexing.
   - **Stop IA**: Halts the index agent. This does not stop the Windows service or the JBoss instance.
   - **Index selected list of objects**: Indexes a set of objects specified by either a DQL statement or a file that contains a list of objects specified by r_object_id.
   - **Logoff**: Logs out of the index agent UI.
   - **Refresh**: Updates statistics and status.

Stopping the index agent

When you stop indexing using the index agent UI, the agent is still running. You must stop the agent using a JBoss script or Windows service.

In the following commands, *Index_agent* is the value you specified for **Server Name** during installation.

Windows:

Stop the index agent service **Documentum *Index_agent*** or run*dsearch_home*\jboss4.3.0\server\stop*Index_agent*.cmd (Windows).

Linux:

Run *dsearch_home*/jboss4.3.0/server/start*Index_agent*.sh (Linux).

Deleting an index agent

Deleting a FAST index agent

1. Log on to the FAST index agent host as the user who installed the FAST index agent.
2. Stop the FAST index agent using Documentum administrator.

3. Start the FAST Index Agent Configuration Program located in $DOCUMENTUM_SHARED/IndexAgents (UNIX/Linux) or **Start > Programs > Documentum > Index Agent Configuration Program** (Windows). Use this program to delete the index agent.

Deleting an xPlore index agent

1. Run *dsearch_home*\setup\indexagent\configIndexagent.bat (Windows) or *dsearch_home*/setup/indexagent/configIndexagent.sh (Linux).
2. Choose the instance to delete.

   Note: If you are unable to connect to the repository, stop all JBoss processes in *dsearch_home*, and then manually delete the JBoss index agent server instance.

Chapter 5

Upgrading xPlore

Requirements — Back up your xPlore customizations including configuration files and stop word lists before upgrade. You later merge those customizations with the upgraded xPlore.

1. Install the latest Content Server hotfix for your version and environment. See Installing xPlore with Content Server 6.5 SPX or 6.6, page 22.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: If you have upgraded Content Server to 6.7, you must restart Content Server before upgrading xPlore.

2. Unzip one of these files to a temporary directory:

   `Documentum_xPlore_1.1_linux_x64.zip` (64–bit)
   `Documentum_xPlore_1.1_linux.zip` (32–bit)
   `Documentum_xPlore_1.1_windows_x64.zip` (64–bit)
   `Documentum_xPlore_1.1_windows.zip` (32–bit)

3. Stop all xPlore instances.
4. Start the primary xPlore instance for upgrade. Do not start any other instances.
5. Run setup.exe (Windows) or setup.bin (Linux) and choose the install directory (the existing xPlore root directory). If you do not install into the existing xPlore root directory, your existing installation is not upgraded. The installer detects your existing xPlore and asks whether you wish to upgrade.

   Note: On 64-bit Windows Server 2008, run these executables on a command line and click **Allow** when prompted.

   You might be required to restart the system.

6. Start the configuration program: *dsearch_home*`\setup\dsearch\configDsearch.bat` (Windows) or *dsearch_home*`/setup/dsearch/configDsearch.sh` (Linux).
7. Choose **Upgrade Existing Server Instance** and then select the instance to upgrade.
8. Provide the installation owner password and optional domain name.
9. Restart the primary instance.
10. Keep your secondary instances stopped while you upgrade them.
11. Restart all instances.
12. Delete and then recreate the index agents. The old index agents do not have the latest upgrade. See Chapter 4, Installing and Configuring an Index Agent.
13. Drop indexes that were created by version 1.0. You cannot perform folder descend queries on these indexes.

    a. Make sure xPlore instances are running.

b. Use the xPlore CLI to run a dropIndex command. See *Documentum xPlore Administration Guide* for information on using the CLIs.

c. Enter the following command:

```
xplore dropIndex 'dftxml', 'folder-list-index'
```

14. Test your upgraded xPlore installation and then migrate your customizations. See Chapter 6, Validating Installation.

15. If you have backups, back up the upgraded xPlore instances.

Caution: You cannot restore an xPlore 1.0 backup, as the xDB version has changed. Back up immediately after upgrading xPlore to 1.1.

Chapter 6

Validating Installation

Validating that you have successfully installed xPlore:

Always test your xPlore installation in multi-user mode before installing it into your production environment.

Starting xPlore Administrator

1. Open your web browser and enter the following:

   `http://`*`host:port`*`/dsearchadmin`

   host:port are specific to your xPlore primary instance.
2. Again enter the host and port for the primary instance and the password for the administrator.

Indexing data

Before uploading test documents to index, install an xPlore index agent, which automatically configures a domain, or manually configure a domain using xPlore Administrator.

1. In the xPlore Administrator, select **Diagnostic and Troubleshooting** > **Upload Testing Document**.
2. Specify the appropriate values in these fields:
   - **Domain** — the domain to which to add the document.
   - **Collection** — the collection to which to add the document.
3. Select one of these options:
   - To upload a document when the xPlore Administrator and the indexing service are on the same host, use **Option 1**, specifying the appropriate values for these fields:
     — **Local File**: Fully qualified path to the file you want to index (click **Browse** to navigate your local file system).
     — **Content Type**: Format of the document

  - **Object ID**: (Optional) String value for the r_object_id element; if this field is not specified the r_object_id value is the concatenation of the file name and a random number.
  - **Owner Name**: (Optional) String value for the r_modifier element
  - **Creation Date**: (Optional) Date that the document was created.
- To upload a document from a shared data store or test a remote CPS instance, choose:
  - **Option 2** and specify the URI, using the file protocol syntax, to the document in the **Remote File** field. All other fields are identical to the ones in **Option 1**.
  - **Option 3**, click **Specify raw XML**, and copy and paste your XML text (for example, XML in the DFTXML format).

  For information about specifying data store locations, see the *xPlore Administration Guide*.

4. Click **Upload**.
5. To verify that no errors have been recorded in the log, select **System Overview** > ***instance*** **(*host*)** > **Logging.**

   where:
   - *instance* is the name of an xPlore instance (for example, PrimaryDsearch is the default name of the primary instance)
   - *host* is the name of the machine on which the xPlore instance is installed
6. To validate that your file has been correctly indexed, submit a query as described in Searching, page 36.

Searching

Before querying the index, install a Documentum index agent for xPlore and index some content, or upload content using xPlore administrator.

1. In the xPlore Administrator, select **Diagnostic and Troubleshooting** > **Test Search**.
2. Perform one of these actions:
   - **Keyword**: Specify the text for which you want to search.
   - **XQuery**: Enter a query using XQuery syntax.
3. Select a domain, collection, language, and maximum number of results and then click **Search**.
4. Click **Search**.
5. Verify that the correct results are displayed.

Verifying that a search executed against xPlore

To check whether a search is executed against FAST or xPlore, use the `ENABLE(FTDQL_DATA)` hint. For example, issue this iAPI command (one a single line):

```
?,c,SELECT r_object_id FROM dm_sysobject SEARCH DOCUMENT CONTAINS 'dmadin' ENABLE(FTDQL_DATA)
```

If the Content Server executes against xPlore, the result contains an XQuery snippet. For example:

```
...Native Query:
[((. ftcontains  ( (('dmadmin') with stemming) ) ))
and ( ( (dmftinternal/i_all_types = '0300271080000105') )
and ( (dmftversions/iscurrent = 'true') ) )]
```

Chapter 7

Migrating from FAST to xPlore

The following tasks describe how to migrate from the FAST indexing server to xPlore:

- Immediately replacing FAST with xPlore, page 39
- Migrating indexes, page 40
- Migrating large environments, page 41
- Troubleshooting migration, page 45
- Enabling FAST with Content Server 6.7, page 46

To test xPlore and FAST in dual mode, see Chapter 8, Testing xPlore and FAST in Dual Mode.

Immediately replacing FAST with xPlore

You can replace FAST by uninstalling FAST, installing xPlore, and reindexing. This process introduces the largest amount of downtime and greatest risk. If hardware availability is an issue, you could index to xPlore (perhaps on the weekend) and bring down FAST before enabling queries.

1. Deploy or upgrade to Content Server 6.5 SP2 or later. Configure at least one Documentum repository and one connection broker. If you upgrade to Content Server 6.7, skip step 5.
2. Install the latest cumulative hotfix for your version of the Documentum Content Server.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: On AIX and Solaris, use GNU tar to untar the cumulative hotfix; otherwise, in the Jboss directory, the file names are truncated.
3. Use Documentum Administrator to stop the Documentum index agent and FAST index server: Navigate to **Administration > Indexing Management > Index Agents and Index Servers**, select the index server or agent, and choose **Tools > Stop**.
4. Remove FAST: Uninstall the FAST indexing server using the instructions in your version. Delete the Documentum index agent for FAST and stop the Content Server. See Deleting an index agent, page 31.
5. Content Server 6.5 SPX or 6.6 only: Run the script fulltext_setup_for_dss.ebs to create objects for xPlore. (Support is already installed with Content Server 6.7.) The script is located in `%DM_HOME%\install\admin` (Windows) or `$DM_HOME/install/admin` (Unix) on the Content Server machine.

```
dmbasic -f fulltext_setup_for_dss.ebs -eSetupForDSS -- repositoryName
username password fulltextHome
```

where:

- *repositoryName*: Name of the repository to set up for xPlore.
- *username* and *password*: Username and password for repository installation owner.
- *fulltextHome*: Fulltext directory on the Content Server, for example, `/export/user1/Documentum/fulltext` (Linux) or `C:\Documentum\fulltext` (Windows).

Note: Enter a space before and after the double hyphen. Do not use environment variables in paths. For example:

```
dmbasic -f fulltext_setup_for_dss.ebs -eSetupForDSS -- testenv
tuser1 mypwd C:\Documentum\fulltext
```

6. Restart the Content Server. After successfully starting up Content Server and the connection broker, wait several minutes (typically 15 minutes) before installing the Documentum index agent.
7. Install the xPlore primary instance and start xPlore. See Installing the primary instance, page 22.
8. Install and configure a Documentum index agent for xPlore and start the index agent UI. See Chapter 4, Installing and Configuring an Index Agent.

   Set up index agent filters after you configure the index agent but before you start reindexing. Filters can exclude cabinets, folders, or object types from being indexed. See *xPlore Administration Guide*.
9. Start the index agent UI and select **Start new reindexing operation**. All dm_sysobject, dm_acl, and dm_group objects are indexed unless they are excluded by a filter.
10. After indexing has completed, you see **Reindexing is completed. Please stop IA and start IA in Normal mode**).

    Verify that the reindexing was completed successfully for all objects by using the ftintegrity tool. See *xPlore Administration Guide*.

Migrating indexes

To migrate indexes from FAST to xPlore, reindex your content using one or more xPlore index agents.

Note: Make sure that you set up index agent filters after you configure the index agent but before you start reindexing. Otherwise, the filters cannot be applied to objects that already been indexed. You can install index agent filters that exclude cabinets, folders, or object types from being indexed. See *xPlore Administration Guide*, "Using the index agent filters" section.

Estimate the amount of time you can spend reindexing your content. xPlore provides guidelines for estimating your reindexing time as well as some best practices. For information on estimate indexing time and tuning indexing, refer to *Documentum xPlore Administration Guide*. Some best practices that you can use to reduce reindexing time are:

- Use multiple CPS instances, which reduces crawling time
- Use multiple xPlore instances, which spreads ingestion across several JVMs or hosts.

Migrating large environments

A large environment has a combination of some of the following characteristics: Tens of millions of documents, thousands of users, 8-15 TB of content spread over several file stores, and an existing FAST index smaller than 600 GB for a single node or smaller than 2 TB spread evenly over multiple nodes.

A large migration typically requires more resources than day-forward indexing support.

Use the following migration best practices:

- Plan indexing disk requirements with the xPlore sizing guide available from Powerlink.
- Do full backups before migration and at points during migration. Reserve disk space for multiple backups.
- Test migration in a non-production environment.
- Supporting a large number of ACLs, page 41
- Adding collections, page 41
- Adding temporary instances, page 42
- Ensuring CPU, disk, and I/O capacity, page 43
- Migrating time-based (recent) content, page 43
- Migrate data to specific collections, page 43

Supporting a large number of ACLs

If your environment has more than 500,000 ACLs, turn off compression for ACLs.

1. Stop all xPlore instances.
2. Edit indexserverconfig.xml. Locate the element dmftinternal/acl_name.
3. Set the value of the compress attribute to *false*.
4. Save the config file and restart the primary instance, then restart all secondary instances.

Adding collections

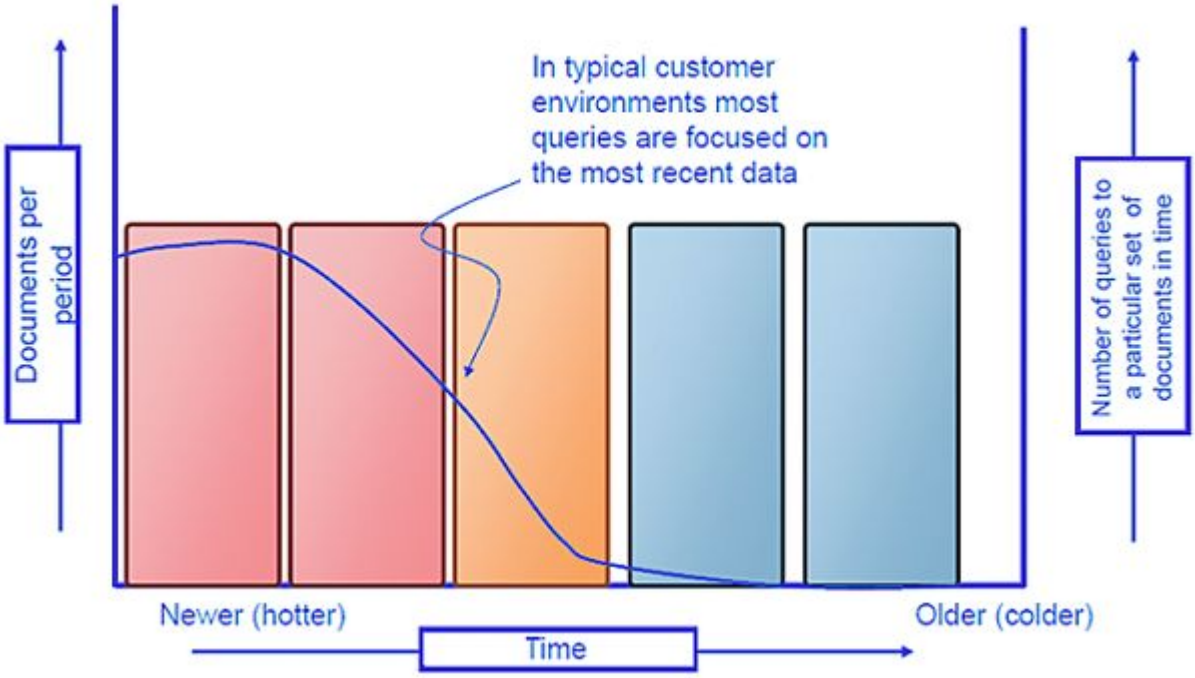
By default, xPlore routes documents in a round-robin fashion over multiple collections. With multiple collections, xPlore leverages concurrent ingestion transactions to different sets of Lucene indexes.

Create at least three backfile collections and ingest documents to those collections. (Add more collections if the hosts have good I/O capacity.) Multiple collections allow parallel inserts into the Lucene indexes. After ingestion, you can consolidate collections using xPlore administrator. You can also configure query partitioning to improve query performance and specificity.

Create at least one day-forward collection to separate old data from new data.

If you use round-robin ingestion, you will not be able to perform bulk deletes of a collection from a specific file store. For large repositories with old data, you can separate old content from new content by collection. Old data is not queried very often, as shown in Figure 3, page 42.

Figure 3. Time-basis of queries



Adding temporary instances

The Content Processing Service (CPS) within xPlore is resource intensive. Each xPlore instance can be configured to have its own CPS. Temporary instances of xPlore with CPS can manage ingestion without impacting ongoing ingestion and queries.

After migration data has been indexed, you can change the collection binding so that all collections are bound to the primary instance. Use xPlore administrator to change collection binding.

1. Create an xPlore secondary instance for each backfile collection created in Adding collections, page 41.
2. Bind each backfile collection to a new temporary CPS instance.
3. Migrate data. See Migrate data to specific collections, page 43.
4. Move backfile collections from the temporary instance to the primary instance. (Change binding using xPlore administrator.)
5. Delete the secondary (temporary) instances.

Ensuring CPU, disk, and I/O capacity

Indexing is both CPU and I/O intensive.

CPU — Extra collections bound to secondary instances, and more cores per instance, meets the temporary increase in CPU requirements.

Disk I/O — I/O is the indexing bottleneck if CPU is sufficient. Use striped allocations for the following areas:

- xPlore data and log directories
- Index agent temporary staging area

Migrating time-based (recent) content

New data can be queried separately from old data, and merges are minimized. To determine whether your query load is primarily for recent content, perform the following steps.

1. Compare results from the following DQL queries of your Content Server:

   ```
   select count(*) from dm_sysobject where datediff(year, r_creation_date,
   r_access_date) < 2 and datediff(year, r_creation_date, r_modify_date) < 2
   ```

   ```
   select count(*) from dm_sysobject
   ```

   The first query counts how many objects were modified and accessed within the past two years. The second query counts the number of objects in the repository. If these counts are close, a high percentage of documents have not been touched after two years. For example, if the first query counts 1 million documents modified and accessed out of 2.5 million objects, 40% of objects are recently touched, and 60% are not of interest to users. Queries for the recent documents can be targeted to age-based collections.

If your queries are time-based, you can create a custom routing class to route older documents to backfile collections. The class routes recent documents and new additions to a day-forward collection. See the example code for this on the EMC Developer Network.

Migrate data to specific collections

1. Define two or three backfile collections using xPlore administrator.
2. Determine the cutoff date for legacy data in these collections.
3. Add routing logic to support the cutoff date. You can add routing logic with one of the following:
   - Custom index agent BOF filter that implements IDfCustomIndexFilter. Base the filter on a date attribute. See "Filtering content from indexing" in *Documentum xPlore Development Guide.*
   - Custom xPlore routing class that tests for creation date, modification date, access date, or custom date attribute for routing to a specific collection. See "Custom routing class" in *Documentum xPlore Development Guide.*

   For an overview of this customization, see Customize routing, page 44.

4. Start backfile ingestion using the index agent UI.
5. Optional: Merge collections for recent data to day-forward (ongoing) collection.

After data has been migrated, new data and updates are directed to a day-forward collection. You can target queries to the recent collection or to all collections. See Target queries to specific collections, page 44.

Customize routing

The Documentum xPlore Development Guide describes routing customization. A custom routing class does the following:

```
If ( obj.r_creation_date < date_cutoff )
{
  Route document to any of the legacy collections
}
else
{
  Route document to day-forward collection
}
```

A custom index agent filter can filter out content based on a date attribute. The filter does the following:

```
If ( obj.r_creation_date < date_cutoff )
{
  Index the document
}
else
{
  Skip the document
}
```

Create and test the routing or filter class before you create the day-forward (ongoing) collection.

Target queries to specific collections

Default queries can be targeted to the most recent collection. Special queries can examine all collections in parallel. Route queries with one of the following:

- DQL: IN COLLECTION clause. For example:

  ```
  select r_object_id from dm_document search document contains 'benchmark'
  in collection('default')
  ```

- DQL hints file: Requires disabling XQuery generation, and does not return faceted results. For example:

  ```
  select r_object_id from dm_document search document contains 'benchmark'
  enable(fds_query_collection_default)
  ```

- DFC QueryBuilder API addPartitionScope()
- DFC IDfXQuery API collection()
- DFS PartitionScope object in a StructuredQuery implementation

For details on creating queries with DFC and DFS APIs, see *Documentum Search Development Guide.*

Required and optional tuning

Implement these requirements and optional tuning parameters for a successful large migration.

Table 4. Ingestion tuning parameters

Parameter	Where	Value
xPlore OS	xPlore host	Required: 64-bit OS
Disk I/O subsystem	xPlore host	Required: Supports at least 1500 I/O per sec. Not required after migration.
xPlore instance JVM	Start script in *dsearch_home*/jboss4.3.0/server	Required: At least 4 GB per xPlore instance.
xDB buffer cache	WEB-INF/classes/indexserver-bootstrap.properties in the primary instance *war* file	Required: Stop all xPlore instances and increase *xhive-cache-pages* to 512 MB - 1 MB (in bytes). Restart xPlore instances.
Number of open files	*nofile* parameter in /etc/security/limits.conf	Required for Linux: Set to 65,000+ and reboot OS. Without it, the xPlore process can run out of file descriptors.
Index agent queue size	indexagent.xml in index agent instance WEB-INF/classes	Required: Set *queue_size* in indexer_plugin_config.indexer to 1500.
Index agent callback queue size	indexagent.xml in index agent instance WEB-INF/classes	Required: Set *callback_queue_size* in indexer_plugin_config.indexer to 1000.
xdb.lucene.ramBufferSizeMB	xdb.properties in primary instance war file dsearch.war/WEB-INF/classes	Default is 3 MB. This can be set to 64 MB or greater. Reset to default after migration.
xdb. lucene.cleanMergeInterval	xdb.properties	Default = 300 sec. Set to 900 for a long migration, resulting in fewer flushes to disk.. Reset to default after migration.
xdb. lucene.finalMergingInterval	xdb.properties	Required: Set to double or triple the default (14400 sec). Reset to default after migration.

Troubleshooting migration

To check the full-text config objects in the Content Server

You have one full-text config object for xPlore, one for FAST, or both.

1. Get the object ID of the dm_ftengine_config object:

   ```
   retrieve,c,dm_ftengine_config
   ```

2. Use the object ID to get the full-text config object parameters. Substitute the object ID returned by the retrieve API.

   ```
   ?,c,select param_name, param_value from dm_ftengine_config where r_object_id=
   <dm_ftengine_config_object_id>
   ```

   Parameters that begin with *dsearch* indicate the xPlore config object.

Enabling FAST with Content Server 6.7

The Documentum 6.7 Content Server does not have support for the FAST indexing server enabled. xPlore is the default indexing server. If you have used FAST and want to continue to use it while you evaluate or migrate to xPlore, use these instructions.

Note: The FAST indexing server is not supported by Microsoft after 2011. xPlore is a fully functional replacement with many added features and better performance.

To enable the FAST indexing server before configuring the Content Server

1. Create a fast.ini file with the following case-sensitive entries:

   ```
   [INDEX AGENT]
   INDEX_AGENT_TYPE=Fast
   ```

2. Launch the Content Server configuration program with the following command for your environment. The Server configuration program is located in in $DOCUMENTUM/product/*version*/install/admin (UNIX and Linux) or %DOCUMENTUM%\product*version*\install\admin (Windows). Substitute the full path to fast.ini or place the file in the launch directory:

   ```
   UNIX or Linux:
   dm_launch_server_config_program.sh -config fast.ini

   Windows:
   Server_Configuration_Program.exe -config fast.ini
   ```

3. If you are upgrading from Content Server 6.5 SP2 or SP3, do the following using iAPI. Substitute your values for FAST_indexserver_hostname and desired FAST_search_base_port (default *13000*):

   ```
   iAPI>retrieve,c,dm_ftengine_config
   iAPI>append,c,l,param_name
   SET>fds_qrserver_host
   iAPI>append,c,l,param_name
   SET>fds_qrserver_port
   iAPI>append,c,l,param_value
   SET>FAST_indexserver_hostname
   iAPI>append,c,l,param_value
   SET>FAST_search_base_port+2100
   iAPI>save,c,l
   ```

To enable the FAST indexing server after configuring the Content Server

1. Run the script fulltext_setup_for_fast.ebs in $DOCUMENTUM/product/6.7/install/admin (UNIX and Linux) or %DOCUMENTUM%\product\6.7\install\admin (Windows).

   ```
   dmbasic -f fulltext_setup_for_fast.ebs -eSetupForFAST --
   repository_name username
   ```

password fulltextHome isRemoteCS

- *repository_name* is the name of the repository against which you are running the consistency checker.
- *username* is the username of a repository superuser.
- *password* is the superuser password.
- *fulltextHome* is the path to the fulltext subdirectory of $DOCUMENTUM or %DOCUMENTUM%.
- *isRemoteCS* specifies whether the Content Server is remote. Not case sensitive. Valid values: t and f.

For example:

```
dmbasic  -f fulltext_setup_for_fast.ebs -eSetupForFAST -- testenv admin N0123
c:\documentum\fulltext f
```

2. Restart the Content Server.
3. If you are upgrading from Content Server 6.5 SP2 or SP3, do the following using iAPI. Substitute your values for FAST_indexserver_hostname and desired FAST_search_base_port (default *13000*):

```
iAPI>retrieve,c,dm_ftengine_config
iAPI>append,c,l,param_name
SET>fds_qrserver_host
iAPI>append,c,l,param_name
SET>fds_qrserver_port
iAPI>append,c,l,param_value
SET>FAST_indexserver_hostname
iAPI>append,c,l,param_value
SET>FAST_search_base_port
iAPI>save,c,l
```

Chapter 8

Testing xPlore and FAST in Dual Mode

To plan your dual-mode testing, see Planning your dual-mode installation, page 49 and

There are two ways to test xPlore and FAST in dual mode:

- Two Content Servers (see Installing FAST and xPlore on two Content Servers, page 50)
- One Content Server (see Installing FAST and xPlore on one Content Server, page 56.

For dual-mode troubleshooting, see Troubleshooting dual mode, page 58.

After you have completed testing, uninstall FAST (see Uninstalling FAST on Content Server 1, page 54 or Uninstalling FAST on one Content Server, page 58).

Planning your dual-mode installation

When you run both xPlore and FAST, there is no downtime, but you require more resources and a more complex process. There are several possible configurations of dual mode:

- FAST on the primary Content Server, xPlore on a secondary Content Server.
- xPlore on the primary Content Server, FAST as backup on the secondary Server.
- Two index servers and index agents, one Content Server. Refer to Installing FAST and xPlore on one Content Server, page 56.

As part of your planning, compare the indexing and search features of xPlore against FAST and validate the entire xPlore system for production use. xPlore and FAST run against the same production repository. This dual mode (or rolling upgrade) is a cost-effective way of minimizing data and environment duplication without adversely affecting your users. You also can execute a planned, incremental migration of users to xPlore. After validating your xPlore installation, you can turn off indexing and search requests and uninstall FAST (see Uninstalling FAST on Content Server 1, page 54).

Note: Dual mode is not supported on Content Server 6.7.

Before installing in dual mode, consider whether your xPlore configuration is set up for high availability (HA). Will both FAST and xPlore fail over? If your FAST backups continue while you install xPlore, then you must budget for the extra hardware. How many users, and in what order, are moved to xPlore? Use the xPlore sizing tool, available in Powerlink, to plan to accommodate the user load.

Installing FAST and xPlore on two Content Servers

Prerequisites — You have an existing installation of the FAST indexing server that is indexing your Content Server version 6.5 SPX or 6.6.

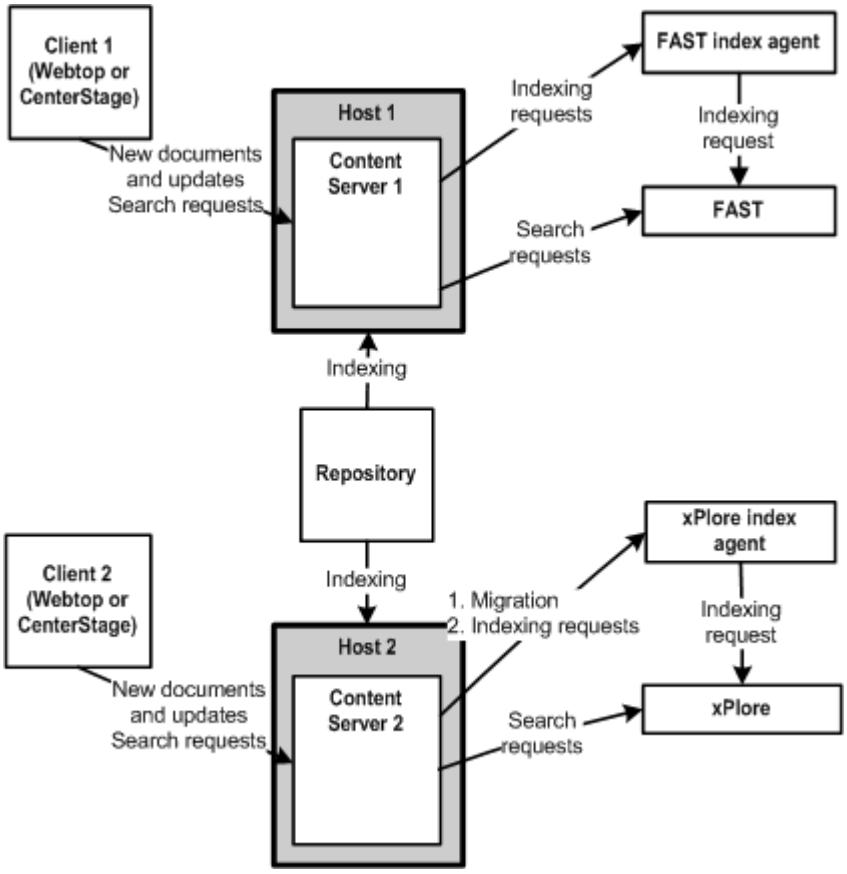
How dual mode works on two Content Servers

Figure 4, page 51 shows a production Content Server with a FAST index agent, a FAST index, and a client such as Webtop or CenterStage. The second host has a Content Server (also known as a remote Content Server) that connects to your production repository. You install an xPlore index agent that indexes all of the production repository existing data into xPlore and then indexes incoming data. You incrementally switch users from the FAST query client to the xPlore query client. Validate the entire system for production use:

- Search experience
- Ease of use
- Administration
- Capacity

After validation, turn off FAST.

Figure 4. FAST and xPlore with two Content Servers



Installing xPlore to Content Server 2

1. Install the latest cumulative hotfix for your version of the Documentum Content Server on Content Server 1 (version 6.5 SP2, SP3, or 6.6). This step is for eventually switching to use xPlore on Content Server 1.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: On AIX and Solaris, use GNU tar to untar the cumulative hotfix; otherwise, in the Jboss directory, the file names are truncated.

2. Install Content Server 2 with the same version as the first, on a different host, and then apply the latest cumulative hotfix. Install a connection broker, but do not install a repository. Filestores must be shared between Content Server 1 and Content Server 2 so that the xPlore index agent can retrieve the data for indexing.

3. Stop Content Server 2.

4. Delete references to machines other than Content Server 2 connection broker in the Content Server 2 file %DOCUMENTUM%\config\dfc.properties (Windows) or $DOCUMENTUM_SHARED/config/dfc.properties (Unix).

5. Delete the Content Server 1 projection entries in the following file on Content Server 2: %DOCUMENTUM%\dba\config*repositoryName*\server_*Server2_repository*.ini (Windows) or $DOCUMENTUM/dba/config/*repositoryName*/server_*Server2_repository*.ini (Unix).

6. Start Content Server 2 and run the script fulltext_setup_for_dss.ebs to create objects for xPlore. The script is located in %DM_HOME%\install\admin (Windows) or $DM_HOME/install/admin (Unix).

   ```
   dmbasic -f fulltext_setup_for_dss.ebs -ePostSetupForFTDualMode -- repository.
   serverConfig superuser password fulltextHome
   ```

   - *serverConfig*: Content Server 2 server config name, which is specified as the value for server_config_name in Content Server 2 server.ini.
   - *fulltextHome*: Full path to the Documentum installation fulltext directory, for example, /export/user1/Documentum/fulltext (Linux) or C:\Documentum\fulltext (Windows)

   Note:

   - Enter a space before and after the double hyphen.
   - No environment variables in paths.

7. You can use the same global registry for both Content Servers. If you use different global registry repositories, project the global repository Content Server to Content Server 2. Add the global registry connection broker to dfc.properties on the Content Server 2 host:

   ```
   dfc.docbroker.port[N]=yourport
   dfc.docbroker.host[N]=yourhost
   ```

   To determine whether a repository is a global registry, use the following iAPI command. If any value in the attribute *docbase_roles* is *Global Registry*, then the repository is a global registry.

   ```
   dump,c,docbaseconfig
   ```

8. Stop and restart Content Server 2. Wait several minutes (typically 15 minutes) for complete startup of Content Server, the connection broker, and xPlore before installing the Documentum index agent.

9. Install the xPlore primary instance on the Content Server 2 host. See Installing the primary instance, page 22.

10. Uncomment the following line in the *dsearch_home*\setup\indexagent\config.properties file:

    ```
    indexagent.dualmode=true
    ```

    This step creates an xPlore full-text user, *dm_fulltext_index_user_01*, and registers this user for full-text events. The name is recorded in indexagent.xml in the index agent JBoss instance. It also sets the *is_standby* property of *dm_fulltext_index* to 1 (true).

11. Install and configure the xPlore index agent for Content Server 2. See Chapter 4, Installing and Configuring an Index Agent.

    Note: Set up index agent filters that exclude cabinets, folders, or object types from being indexed. Configure your filters before you start indexing. Filters cannot be applied to objects that already been indexed. See "Configuring index agent filters" in *xPlore Administration Guide*.

12. Retrieve the xPlore *FTENGINEID* for the next step, using this iAPI query:

    ```
    ?,c,select ft_engine_id from dm_fulltext_index where is_standby=1
    ```

13. Specify xPlore as the full-text engine in the Content Server 2 file %DOCUMENTUM%\dba\config*repository*\server_*repository*.ini (Windows) or

$DOCUMENTUM/dba/config/*repository*/server_*repository*.ini (Unix). Add the following line at the end of the [SERVER_STARTUP] section:

```
ftengine_to_use = FTENGINEID
```

14. Stop and restart or reinitialize Content Server 2.
15. Start the xPlore index agent. See Starting the index agent, page 30. Choose **Start new reindexing operation** to index all repository content and dm_sysobject, dm_acl, and dm_group objects.
16. After indexing is completed, the index agent UI displays **Reindexing is completed. Please stop IA and start IA in Normal mode**. Verify that the reindexing was successful for all objects by using the ftintegrity tool. See "Verifying index migration with ftintegrity" in *xPlore Administration Guide*.

Configuring Content Server 2 for search

Use Content Server 2 for all search requests (SEARCH DOCUMENT CONTAINS queries) from clients such as IAPI, IDQL, Webtop, and CenterStage. A second client (Webtop or CenterStage) connects to Content Server 2 to execute queries against xPlore.

Note: Content Server 2 does not need to access the production Content Server local content store directly. With two Content Servers, the local content store is shared, which can negatively affect performance. After indexing (migration) has completed on Content Server 2 , you can configure this Server for search requests only. The index agent for xPlore connects (through the corresponding connection broker) to the production Content Server instead of Content Server 2. All content access and indexing requests are performed on the production Content Server only.

1. Create a distributed store so that each Content Server has a local store.
2. Change the default store in the application type to the name of the distributed store.
3. Before you decommission Content Server 2, replicate all of the content in its local component store to the local component store of the production Content Server.

Verifying dual mode

1. Verify that there are two of the following objects: dm_ftindex_agent_config, dm_ftengine_config, dm_fulltext_index, and dm_fulltext_index_user_N (N is null or 1). For example:

```
?,c,select r_object_id,object_name from dm_ftindex_agent_config
?,c,select r_object_id,object_name from dm_ftengine_config
?,c,select r_object_id,index_name from dm_fulltext_index
?,c,select r_object_id,user_name from dm_user where user_name like
'dm_fulltext_index%'
```

2. Verify that the FAST indexing engine is installed:

```
retrieve,c,dm_ftengine_config where object_name like '%FAST%'
```

3. Verify that two fulltext index users are generating queue item events. For example, execute an API to check the dmi_registry table:

```
?,c,select registered_id,user_name from dmi_registry where
user_name like 'dm_fulltext_index_user%'
```

4. Verify that two queue items are generated for an object:
   1. Stop both index agents to prevent queue item processing.
   2. Create an object of type dm_document and save it.
   3. Get the object IDof the new object.
   4. Run the following command to see the two queue items created for this object. Replace XXX with the object ID of the new object:

      ```
      ?,c,select r_object_id,name from dmi_queue_item where item_id='XXX'
      ```
5. Verify that documents are indexed.
6. Verify that xPlore processes a query using iAPI, iDQL, Webtop or CenterStage.

Uninstalling FAST on Content Server 1

1. If you have not already done so, install the latest cumulative hotfix for your version of Content Server 1.

   Caution: If you do not install the hotfix, xPlore does not work properly.

   Note: On AIX and Solaris, use GNU tar to untar the cumulative hotfix; otherwise, in the Jboss directory, the file names are truncated.
2. Execute %DM_HOME%\install\admin\fulltext_setup_for_dss.ebs (Windows) or $DM_HOME/install/admin/fulltext_setup_for_dss.ebs (UNIX/Linux) on Content Server 1:

   ```
   dmbasic -f fulltext_setup_for_dss.ebs -eConsolidateFulltextEngine -- repository_name
   superuser_name superuser_password fulltext_install_location
   ```

   Note:
   - Enter a space before and after the double hyphen.
   - No environment variables in paths.

   For example:

   ```
   dmbasic -f fulltext_setup_for_dss.ebs -eConsolidateFulltextEngine -- dual_mode_repo
   dmadmin password C:\Documentum\fulltext
   ```
3. Shut down Content Server 1, Content Server 2, and both index agents.
4. Delete the FAST index agent. See Deleting an index agent, page 31.
5. On the xPlore index agent host, add Content Server 1 to dfc.properties. This file is located in *dsearch_home*/jboss4.3.0/server/DctmServer_Indexagent/deploy/IndexAgent.war/WEB-INF/classes.

   ```
   dfc.docbroker.port[N]=yourport
   dfc.docbroker.host[N]=yourhost
   ```
6. Unregister the FAST queue user for full-text events. The first iAPI command returns two queue_user objects. The xPlore user has a port number in the object name, the FAST user does not. The user is also identified in indexagent.xml in the index agent JBoss instance.

   ```
   API>?,c,select object_name, queue_user from dm_ftindex_agent_config
   ```

Substitute the queue_user returned by this command for the user_name in the next comand. The user name must be within single quotes:

```
API>execsql,c,delete dmi_registry_s where user_name='XXX'
```

7. Delete the FAST queue user (dm_fulltext_index_user). The user name must be within single quotes:

   ```
   API>?,c,delete dm_user object where object_name = 'XXX'
   ```

8. Delete the FAST full-text engine config object (dm_ftengine_config) and the standby dm_fulltext_index object. Using iAPI, retrieve the dm_ftengine_config object. For example:

   ```
   API>?,c,select r_object_id,object_name from dm_ftengine_config
   r_object_id       object_name
   --------------------------------
   080019c880000d17  FAST Fulltext Engine Configuration

   API>?,c,delete dm_ftengine_config object where r_object_id = '080019c880000d17'
   ```

9. Delete the FAST fulltext location object on Content Server 1. First get the object ID, for example:

   ```
   ?,c,select r_object_id from dm_location where object_name='fast'

   r_object_id
   ----------------
   3a00031580000154
   ```

   Delete the object:

   ```
   API> destroy,c,3a00031580000154
   ```

10. Delete Content Server 2 and its connection broker. Refer to the *EMC Documentum Content Server Installation Guide*.

Using xPlore on Content Server 1

1. Install xPlore on Content Server 1. See Installing xPlore with Content Server 6.5 SPX or 6.6, page 22.
2. Set the value of *dm_server_config.fulltext_location* of the Content Server to *dsearch*, like the following iAPI commands:

   ```
   retrieve,c,dm_server_config
   set,c,l,fulltext_location
   dsearch
   save,c,l
   ```

3. Get the ID of the dm_ftengine_config object to use in the next step:

   ```
   retrieve,c,dm_ftengine_config where object_name like '%DSearch%'
   ```

4. Edit the Content Server server.ini file located in *documentum_home*/dba/config/*servername*. Add a [SERVER_STARTUP] property *ftengine_to_use* and set it to the ID returned by the previous step.
5. Restart the Content Server and xPlore index agent and check the Content Server log to make sure the dsearch query plugin is loaded. For example, from *Documentum_home*/dba/log/*domain_name*.log:

   ```
   Loaded FT Query Plugin: C:\Documentum\product\6.6/bin/DSEARCHQueryPlugin.dll
   ```

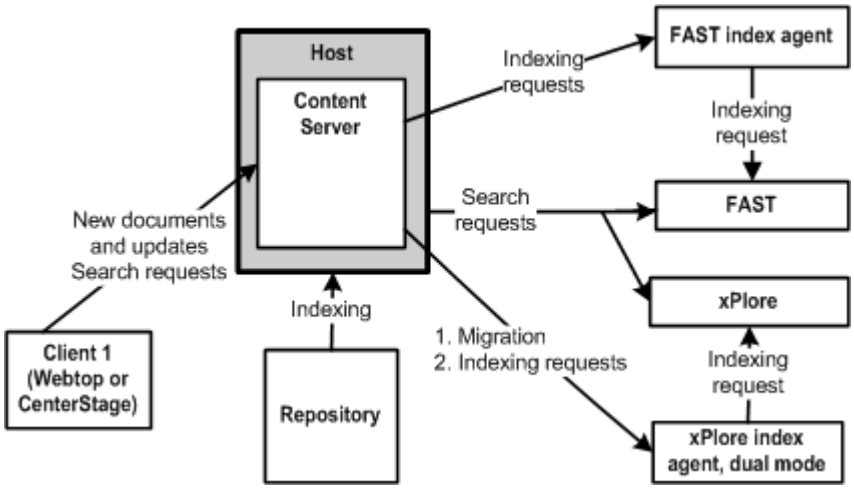
Installing FAST and xPlore on one Content Server

Prerequisites — You have an existing installation of the FAST indexing server that is indexing your Content Server version 6.5 SPX or 6.6. Dual mode with FAST and xPlore with Content Server 6.7 is not supported.

How dual mode works on one Content Server

Figure 5, page 56 shows an installation of xPlore and an xPlore index agent to provide indexing in parallel with your FAST installation. You can switch the search server from FAST to xPlore to evaluate search. The following procedure describes how to set up one Content Server to switch back and forth between FAST and xPlore.

Figure 5. FAST and xPlore with one Content Server



Installing xPlore to a FAST Content Server

1. Shut down the Content Server and apply the latest patch.
2. Start the Content Server.
3. Execute the method SetupDualModeForSingleCS in the script fulltext_setup_for_dss.ebs. This file is located in `%DM_HOME%\install\admin` (Windows) or `$DM_HOME/install/admin` (Unix).

```
dmbasic -f fulltext_setup_for_dss.ebs -eSetupDualModeForSingleCS -- repositoryName
   username password fulltextHome fulltextUserN
```

where:

- *repositoryName*: Name of the repository to set up for xPlore.
- *username* and *password*: Username and password for repository installation owner.
- *fulltextHome*: Fulltext directory on the Content Server, for example, `/export/user1/Documentum/fulltext` (Linux) or `C:\Documentum\fulltext` (Windows).
- *fulltextUserN*: Full-text user for xPlore, with an integer appended. (FAST user is dm_fulltext_index_user.)

Note: Enter a space before and after the double hyphen. Do not use environment variables in paths.

For example:

```
dmbasic -f fulltext_setup_for_dss.ebs -eSetupDualModeForSingleCS -- testenv
tuser1 mypwd C:\Documentum\fulltext dm_fulltext_index_user_01
```

4. Stop and restart the Content Server.
5. Install and configure xPlore. See Installing xPlore with Content Server 6.5 SPX or 6.6, page 22.
6. Start the xPlore primary instance:
   - Windows: The **Documentum Search Services *PrimaryDsearch* (Primary)** Windows service, or *dsearch_home*\jboss4.3.0\server\startPrimaryDsearch.cmd
   - Linux: *dsearch_home*/jboss4.3.0/server/startPrimaryDsearch.sh
7. Install and configure an xPlore index agent. See Chapter 4, Installing and Configuring an Index Agent.
8. Follow the procedure in Switching from FAST to xPlore, page 57 to begin testing xPlore.
9. To switch to FAST, see Switching from xPlore to FAST, page 58.
10. After validation of xPlore, uninstall FAST (Uninstalling FAST on one Content Server, page 58).

Switching from FAST to xPlore

1. Set the value of *dm_server_config.fulltext_location* of the Content Server to *dsearch*, like the following iAPI commands:

   ```
   retrieve,c,dm_server_config
   set,c,l,fulltext_location
   dsearch
   ```

2. Get the ID of the dm_ftengine_config object to use in the next step:

   ```
   retrieve,c,dm_ftengine_config where object_name like '%DSearch%'
   ```

3. Edit the Content Server server.ini file located in *documentum_home*/dba/config/*servername*. Add a [SERVER_STARTUP] property *ftengine_to_use* and set it to the ID returned by the previous step.
4. Restart the Content Server and xPlore index agent and check the Content Server log to make sure the dsearch query plugin is loaded. For example, from *Documentum_home*/dba/log/*domain_name*.log:

```
Loaded FT Query Plugin: C:\Documentum\product\6.6/bin/DSEARCHQueryPlugin.dll
.
```

Switching from xPlore to FAST

1. Set the value of *dm_server_config.fulltext_location* of the Content Server to *fast*, like the following iAPI commands:

   ```
   retrieve,c,dm_server_config
   set,c,l,fulltext_location
   fast
   save,c,l
   ```

2. Get the ID of the dm_ftengine_config object to use in the next step:

   ```
   retrieve,c,dm_ftengine_config where object_name like '%FAST%'
   ```

3. Edit the Content Server server.ini file located in *documentum_home*/dba/config/*servername*. Add a [SERVER_STARTUP] property *ftengine_to_use* and set it to the ID returned by the previous step.

4. Restart the Content Server and xPlore index agent.

Uninstalling FAST on one Content Server

Prerequisites — Switch from FAST to xPlore before uninstalling FAST. See Switching from FAST to xPlore, page 57.

1. Uninstall FAST using the instructions in your full-text installation guide for FAST.

2. Delete the index agent for FAST. See Deleting an index agent, page 31.

3. Unregister the FAST queue user for full-text events. The first iAPI command returns two queue_user objects. The xPlore user has a port number in the object name, the FAST user does not. The user is also identified in indexagent.xml in the index agent JBoss instance. The user name must be within single quotes:

   ```
   API>?,c,select object_name, queue_user from dm_ftindex_agent_config
   API>execsql,c,delete dmi_registry_s where user_name=queue_user
   ```

4. Delete the FAST queue user (dm_fulltext_index_user). Specify the FAST queue_user returned in the previous step. The user name must be within single quotes:

   ```
   API>?,c,delete dm_user object where object_name = queue_user
   ```

Troubleshooting dual mode

FAST not working — You see the following warning in the Content Server log:

```
"A fulltext library is currently loaded. Changes made to load a new fulltext
library cannot take effect until Content Server is restarted."
```

You must enable FAST. See Enabling FAST with Content Server 6.7, page 46.

Checking the FAST log for query information — The log is located in $DOCUMENTUM/fulltext/indexserver/var/log/querylogs.

Chapter 9

Troubleshooting

The following topics describe troubleshooting steps for installation problems:

For troubleshooting xPlore operations, refer to *Documentum xPlore Administration Guide*.

Verifying Content Server support in 6.5 SPX or 6.6

To check whether the Content Server has been enabled for xPlore, check the Content Server log. The xPlore plugin binary should be loaded with an entry like the following (Windows):

```
Loaded FT Query Plugin: C:\Documentum\product\6.6/bin/DSEARCHQueryPlugin.dll
```

To add xPlore support, apply the latest Content Server hotfix (patch) for your environment.

Caution: If you do not install the hotfix, xPlore does not work properly.

If you have installed the hotfix, do the following:

1. Using Documentum Administrator, log in as a user with superuser privileges.
2. Execute the following DQL query to verify that Content Server has the correct full-text plugin installed:

   ```
   select r_object_id,object_name from dm_ftengine_config
   ```

   Dual-mode configuration shows two object IDs. Single mode configuration shows only one. One mode must show an object with the name *DSearch Fulltext Engine Configuration*.

Cannot create a secondary instance

Investigate the following possible causes:

- The primary instance is not running.
- You specified the wrong host name or port number for the primary instance.
- The configuration directory containing the bootstrap file is not accessible from the secondary instance host.

Can’t start up a secondary instance

Investigate the following possible causes:

- The primary instance is not running.
- A collection that is bound to the instance is corrupted. The xPlore log contains X-Hive exception: DATA_CORRUPTION.

  Workaround: Set the value of *force-restart-xdb* in indexserver-bootstrap.properties to *true* and then restart the instance. The corrupted collection will be marked as unusable.
- (Linux) The mandatory nfslock service is not running. You see the following error:

  ```
  IOException: No locks available
  ```

xPlore administrator problems with IE7 and IE8 browsers

Modify Internet Explorer Enhanced Security in the browser to use xPlore administrator on IE 7 or IE 8.

To configure on Windows 2008:

1. Close all instances of Internet Explorer 7.
2. Click **Start > Administrative Tools > Server Manager**.

   If a User Account Control dialog box appears, click Continue.
3. Under Security Summary, click **Configure IE ESC**.
4. Under Administrators, click OFF.
5. Under Users, click OFF.
6. Click OK and restart the browser.
7. Clear **Tools > Browsing history**.

To configure on Windows 2003:

1. Close all instances of Internet Explorer 7.
2. Open **Control Panel > Add or Remove Programs > Add/Remove Windows Components** (on left side).

3. Choose **Internet Explorer Enhanced Security Configuration**, click the check boxes to clear them, and then click **OK**.
4. Click **Next** and then click **Finish**.
5. Restart Internet Explorer.
6. Clear **Tools > Browsing history**.

To configure Internet Explorer 8:

1. Open **Tools > Internet Options > Security > Trusted Sites**.
2. Add the xPlore administrator URL to the trusted sites and set security level to **Medium-low**.
3. Close all instances of IE and restart IE.

Index agent issues

Global registry not available

Follow these troubleshooting steps in the order noted.

1. Restart global registry Content Server and connection broker.
2. Verify that the global registry repository projects to the same connection broker as the Content Server you are indexing. If not, edit dfc.properties on the index agent host to point to the global registry:

   ```
   dfc.docbroker.port[N]=yourport
   dfc.docbroker.host[N]=yourhost
   ```

3. Verify that the global registry Content Server is a global registry. Use the following iAPI command:

   ```
   dump,c,docbaseconfig
   ```

   If any value in the repeating attribute *docbase_roles* is *Global Registry*, the repository is a global registry repository.
4. If the Content Server is not a global registry, designate it as one. Set the attribute docbase_roles[0] = 'Global Registry' in the docbaseconfig object.

Index agent configuration fails

If configuration fails when you run configIndexagent.bat or configIndexagent.sh, check Indexagent_Configurator_InstallLog.log for errors. Run the script again with a console (on Windows, control-click the batch file) and see a more detailed error description.

Domain or host name cannot contain underscore

The HTTP client specification does not permit underscores in domain or host names. For example, a host name of win2k3_64 will cause the index agent to hang.

Workaround: Use the IP address, or "localhost" if applicable:

1. Stop xPlore and index agent instances.
2. Fetch dm_ftengine_config for xPlore.
3. Change the value of dsearch_qrserver_host to the IP address. For example, using iapi32, execute the following commands:

   ```
   retrieve,c,dm_ftengine_config
   set,c,l,param_value[2]
   SET>IP_address
   save,c,l
   ```

4. Edit indexserverconfig.xml, which is located in *dsearch_home*/config.
5. Locate the node element for the primary node. Change the url attribute value to the IP address.
6. Save the configuration and restart xPlore instances.

Linux installation errors

- Linux: “Too many files open” error (SRCH-424), page 64

Linux: “Too many files open” error (SRCH-424)

On Linux, if an error occurs that states that too many files are open, set the appropriate values in `/etc/security/limits.conf`. For example:

```
soft nofile 1024
hard nofile 1048576
```

This error can occur when the index is under a heavy load.

Linux: `libidn.so.11` file missing

On Linux, indexing fails if the `libidn.so.11` file is missing.

Chapter 10

Uninstalling xPlore

To completely uninstall an xPlore installation, repeat the following procedure for every machine in your xPlore federation.

To uninstall xPlore on each host:

1. Stop the index agent: The **Documentum *Index_agent*** Windows service, or *dsearch_home*\jboss4.3.0\server\stop*Index_agent*.cmd (Windows) or *dsearch_home*/jboss4.3.0/server/stop*Index_agent*.sh (Linux). *Index_agent* is the name of the index agent.
2. Stop the primary xPlore instance: Stop the **Documentum Search Services *PrimaryDsearch* (Primary)** Windows service, or run *dsearch_home*\jboss4.3.0\server\stop*PrimaryDsearch*.cmd (Windows) or run *dsearch_home*/jboss4.3.0/server/stop*PrimaryDsearch*.sh (Linux). *PrimaryDsearch* is the name of your primary xPlore instance.
3. Stop secondary xPlore and CPS instances: **Documentum Search Services *Secondary_instance* (Secondary)** Windows service, or run *dsearch_home*\jboss4.3.0\server\stop*Secondary_instance*.cmd (Windows) or run *dsearch_home*\jboss4.3.0\server\stop*Secondary_instance*.sh (Linux).
4. Delete all xPlore instances using the xPlore configuration program on each instance: *dsearch_home*\setup\dsearch\configdsearch.bat (Windows) or *dsearch_home*/setup/dsearch/configdsearch.sh (Linux). Choose **Delete Existing Server Instance**.
5. Delete all index agent instances. Run `dsearch_home\setup\indexagent\configIndexagent.bat` (Windows) or `dsearch_home/setup/indexagent/configIndexagent.sh` (Linux). Choose **Delete existing index agent**.
6. Double-click `dsearch_home/Uninstall/uninstall.exe` and follow the instructions.

Note: If you are unable to connect to the repository, stop all JBoss processes in *dsearch_home* and then manually delete the jboss directory.

If you ran `dsearch_home/Uninstall/uninstall.exe` before deleting all xPlore and index agent instances

1. Set the Windows services for the xPlore and index agent instances to manual.

2. Reboot the machine and delete the *dsearch_home*.

Appendix A

xPlore ports

These topics are included:Table 5, page 67 shows the ports that each xPlore component uses.

Table 5. xPlore ports

xPlore Component	Ports Used
xPlore instance	Base port and the next 100 consecutive ports in ascending order.
xDB	The port (listener) specified by adding 30 to the base port.
xPlore Administrator	The port (RMI) specified by adding 31 to the base port.
CPS	The port specified by adding 22 to the base port.
Index agent	Base port and the next 20 consecutive ports in ascending order.

Index

S

T

U

X