



OpenText™ Documentum™ xPlore

Installation Guide

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

EDCSRC220100-IGD-EN-01

OpenText™ Documentum™ xPlore
Installation Guide
EDCSRC220100-IGD-EN-01
Rev.: 2022-Jan-27

This documentation has been created for OpenText™ Documentum™ xPlore CE 22.1.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product,
on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: <https://support.opentext.com>
For more information, visit <https://www.opentext.com>

Copyright © 2022 Open Text. All Rights Reserved.

Trademarks owned by Open Text.

One or more patents may cover this product. For more information, please visit <https://www.opentext.com/patents>.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However,
Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the
accuracy of this publication.

Table of Contents

1	Planning the xPlore Environment	7
1.1	How Documentum xPlore works	7
1.2	Planning your installation	8
1.3	Determining your requirements	8
1.4	xPlore sizing calculation	9
1.5	Sizing CPU	10
1.6	Sizing memory	10
1.7	Sizing storage	11
1.8	Network and virtual machines	12
1.9	Hardware tradeoffs	13
2	Planning High Availability and Disaster Recovery	15
2.1	About high availability and disaster recovery	15
2.2	Planning for high availability	17
2.3	Two primary instances HA (active-active) process	19
2.4	Planning for disaster recovery	21
2.5	Disaster recovery in a Documentum environment	22
3	Installing xPlore	25
3.1	Before you install	25
3.2	Installing the xPlore primary instance with Documentum Server 7.3 and later	26
3.3	Validating installation	28
3.3.1	Starting xPlore Administrator	29
3.3.2	Testing indexing	29
3.3.3	Testing search	30
3.4	Sharing xPlore directories from the primary instance host	31
3.5	Installing a secondary or spare instance	33
3.6	Configuring SSL connections	36
3.6.1	Configuring SSL connections using script (recommended)	36
3.6.2	Configuring SSL for remote CPS	39
3.7	Adding CPS Instances	41
3.8	Configuring CPS dedicated to indexing or search	43
3.9	Deleting an xPlore or CPS instance	43
4	Installing and Configuring Index Agents	45
4.1	Preparing index agent installation	45
4.2	Configuring the index agent	45
4.3	Starting the index agent	47
4.4	Stopping the index agent	48
4.5	Silent index agent startup	49

4.6	Deleting an index agent	52
5	Deploying xPlore on Docker Compose Environment	53
5.1	Overview	53
5.2	Prerequisites	53
5.3	Deploy xPlore Dockers using Docker Compose	53
5.3.1	Docker Compose Yaml Sample	53
5.3.2	Start and Stop Docker Compose	54
5.3.2.1	Start Docker Compose	54
5.3.2.2	Stop Docker Compose	55
5.3.3	Configure SSL	55
5.4	Environment Parameters	55
5.4.1	IndexServer	55
5.4.2	IndexAgent	57
5.4.3	CPS	61
5.5	Limitations	64
6	Configuring WatchDog	65
6.1	Configuring DM_TICKET in WatchDog	65
7	Performing Silent Installation	67
7.1	Using response file to perform silent installation	67
7.2	Setting up silent installation	67
7.3	Configuring silent installation	69
7.4	Running the silent install script	71
7.5	Verifying silent installation	74
7.6	Troubleshooting silent installation	74
7.7	Sample single-instance installation	75
8	Setting up High Availability	77
8.1	Spare instance HA	77
8.2	Configuring active-active load balancer support	79
8.3	Setting up primary instance HA (active-active)	82
8.4	Enabling queries after active-active failover	83
8.5	Restoring data after a deployment fails	84
8.6	Documentum Server full-text objects and initialization files	85
9	Upgrading xPlore	87
9.1	Upgrade overview	87
9.2	Performing pre-upgrade tasks	87
9.3	Performing an in-place xPlore upgrade	89
9.4	Performing post-upgrade tasks	90
9.5	Upgrading xPlore active-active HA	91
9.5.1	Performing an in-place xPlore HA upgrade	91

10	Migrating from FAST to xPlore	93
10.1	Immediately replacing FAST with xPlore	93
10.2	Migrating indexes (reindexing)	94
10.3	Migrating large environments	95
10.3.1	Large migrations best practices	95
10.3.2	Supporting a large number of ACLs	95
10.3.3	Adding temporary instances	95
10.3.4	Ensuring CPU, disk, and I/O capacity	96
10.3.5	Migrating data to specific collections	96
10.3.6	Migrating content based on document age	97
10.3.7	Custom routing	98
10.3.8	Targeting queries to specific collections	98
10.3.9	Tuning parameters for large migrations	99
10.4	Troubleshooting migration	100
11	Troubleshooting Installation	101
11.1	Debugging installation	101
11.2	Fixing indexserverconfig upgrade issues	102
11.3	Host name contains an underscore	103
11.4	Instance name fails	103
11.5	Cannot create a secondary instance	103
11.6	Cannot start a secondary instance	104
11.7	CPS instance does not start	104
11.8	xPlore administrator problems with Internet Explorer 8 and Internet Explorer 9 browsers	104
11.9	Index agent errors	105
11.9.1	Global registry not available	105
11.9.2	Index agent configuration fails	105
11.9.3	Index agent version and OS	106
11.10	Linux errors	106
12	Uninstalling xPlore	107
12.1	Uninstalling xPlore	107
13	References	109
13.1	xPlore ports	109
13.2	HTTP status codes	109

Chapter 1

Planning the xPlore Environment

This document is intended for system administrators responsible for installing and configuring Documentum xPlore. To use this document, you need the following:

- Administrative privileges on the computer where you are installing Documentum xPlore
- Working knowledge of Microsoft Windows or Linux
- Knowledge of the business nature of the data to be indexed by Documentum xPlore

1.1 How Documentum xPlore works

Documentum 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 Documentum xPlore index service and search service are installed as a WAR file to a WildFly application server that is included in the Documentum xPlore installer. The administration console and online help are installed as war files in the same WildFly application server. The index is stored in the directory *xplore_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 instances and data management services)
- Spare - A spare instance can be manually activated to take over for a disabled instance.



Note: When planning a multiple instance deployment on the same host, you must ensure that the host has sufficient resources for the services that will be deployed on it.

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.

1.2 Planning your installation

1. Determine your sizing and performance requirements.
2. Determine your hardware requirements including xPlore server CPUs and memory, disk I/O, storage, network, and virtual machine requirements.
3. Determine your backup and recovery, high availability, and disaster recovery requirements. See “[Planning for high availability](#)” on page 17 and “[Planning for disaster recovery](#)” on page 21.

1.3 Determining your requirements

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

- 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 and indexed 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 impacts 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 (fewer 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 Documentum 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.

1.4 xPlore sizing calculation

OpenText provides tools and guides for system sizing to help you determine the appropriate number of instances and system capabilities that you require. OpenText Global Technical Services (or qualified third-party integrators) can assist you in reviewing your needs and completing the xPlore sizing tool input.



Note: 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 tool.
- 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.

1.5 Sizing CPU

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

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 4 GB of memory.

1.6 Sizing 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.

The following diagram 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 page server cache, and optional virtual machine working memory. The operating system file buffer cache reserves memory space for temporary files. This cache is dynamic and expands to fill any remaining memory without interfering with xPlore processes.

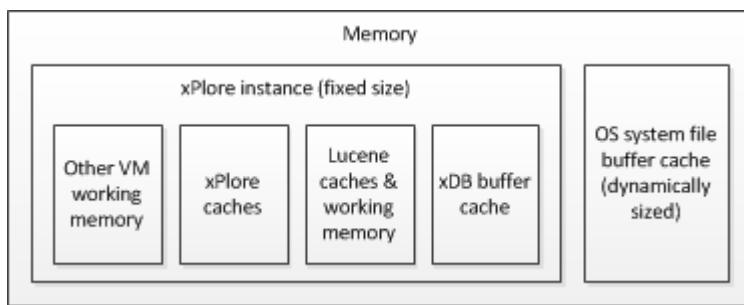


Figure 1-1: xPlore memory usage

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

Table 1-1: 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 page server cache	Used to store XML file blocks (ingest and query).	Increase size for higher query rates.
Operating system file buffer cache	Temporary file. 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.

1.7 Sizing 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.

The following table describes storages areas that are used by xPlore.

Table 1-2: Storage areas

Files	Description	Indexing	Queries
xDB segments	Stores DFTXML, Lucene indexes, and other xDB collections (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 federation logs1	Stores transaction information	Transactions are logged before data is written to disk	When snapshot information is required
Lucene indexes	(in xDB segments) Query lookup and retrieval of facets and security	Index information updated through inserts and merges	Inverted index lookups and facet and security retrieval
Index agent content staging area2	Temporarily stores content during indexing process	Temporarily stores content during indexing process	N/A

1. Old xDB federation logs are purged at each backup.
2. For the index agent content staging area, it is recommended that you allocate disk space with at least an approximate equivalent of the total combined size of the top 2000 largest objects or 2000 times the average content size when batch size is 1000.

1.8 Network and virtual machines

Network

- 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 or 10 GB Ethernet are 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.

1.9 Hardware tradeoffs

The following table shows 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 1-3: Hardware budget: Performance and size

Environment or configuration	Storage	Indexing performance	Query performance	Resources used
Document size	+	+	x	Disk space, memory
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
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 2

Planning High Availability and Disaster Recovery

2.1 About high availability and disaster recovery

Business continuity is the umbrella term that covers all efforts to keep critical data and applications running despite any type of interruption. Interruptions can be both planned and unplanned. Planned interruptions include regular maintenance or upgrades. Unplanned interruptions could include hardware or software failures, data corruption, natural or man-made disasters, viruses, or human error.

Enterprises can use the following solutions to meet business continuity requirements:

- Backup and recovery is an essential part of both basic data disaster recovery and operational recovery strategies. Operational recovery is the ability to recover from errors that can occur on a regular basis but are not catastrophic, for example, data corruption or accidentally deleted files. In data disaster recovery, data backups are replicated but no target hardware/software installation exists. The installation hardware and software must be reinstalled, and then the data backups are reapplied. For information on the types of backup and restore, see *OpenText Documentum xPlore Administration and Development Guide*.
- High availability provides a high level of business continuity as well as high performance through load balancing for mission critical systems. In a high availability configuration, a secondary system is maintained as an exact copy to which the primary system can fail over.
- Disaster recovery responds to catastrophic failures. When your installation fails, you recover it by restoring it to a previously consistent state (that is, a particular point in time) from your backups. To restore your installation to a consistent state, use backup coordinator software like EMC® Networker®. This software ensures that all of your components and data source backups are synchronized.

Your business users and information technology department together can provide a direction for your high availability and disaster recovery planning. For instance, your information technology department can standardize on Cisco or Microsoft clustering.

Recovery point and recovery time (RPO and RTO)

RPO: Decide how much data loss you are willing to incur. Use this decision to calculate how often you need to perform backups. Backups should be performed at fixed intervals. The length of time between backups is called the Recovery Point Objective; that is, the maximum amount of data that you are willing to lose.

When your RPO is in units of minutes or hours, then you will probably need to have a high availability configuration. Because the amount of data lost on the

Documentum Server is reflected in the corresponding xPlore indexes, the xPlore RPO is always the same as the Documentum Server RPO. Because any index data loss (because of a failure or data corruption) can always be recovered by refeeding documents to xPlore, any backup of or high availability for xPlore shortens RTO (not the RPO).

RTO: Decide how long you're willing to wait until the data is completely restored and business applications become available. The time it takes to completely restore data and for business applications to become available is called the Recovery Time Objective. Your RTO can be different from your RPO. For example, you might need an extremely short RPO to minimize potential data loss, but you can have a long RTO in which you can tolerate 12 to 24 hours to recover data and get your business applications back online.

After you determine your RPO and RTO, you can determine how much time you actually have to perform your backups (backup window). The backup window determines the type and level of your backups. For example, if you have a system that requires 24-hour, 7-days-a-week, 365-days-a-year availability, then there is no offline backup window. So, you would have to perform an online backup (also known as a hot backup) in which the system is not taken offline.

To determine your backup and recovery, high availability, and multi-site disaster recovery plan, you first determine the degree to which your xPlore installation is mission critical—that is, if your organization considers your xPlore installation to be down for a short period of time (1 hour for example) as being a severe outage. Your Documentum repository could be mission critical but xPlore full-text search functionality might not be. In general, the shorter your RTO and smaller your RPO, the more mission critical your xPlore installation is.

If you have a short RTO and small RPO, your mission critical installation be designed to recover from any kind of planned or unplanned interruption within the specified RTO and RPO. In general, a mission critical configuration will cost more (in terms of both cost and staffing) than a non-mission configuration because it will require more hardware, planning, and testing. Mission critical systems should not rely solely on backups: You should also consider investing in high availability solutions.

Sometimes a partial recovery that makes your most critical indexes available for querying or indexing new content may be suitable for your xPlore installation. Some examples of an acceptable partial recovery are:

- To make the previous 2 days of indexed data available for querying within 1 hour but make all of your indexes available within 24 hours.
- To make querying of all indexes available within 1 hour but resume indexing within 24 hours.



Note: Partial refeeds and indexing are normally required to restore xPlore to a point-in-time that is synchronized with Documentum Server. There is usually a

lag time between new content that was added to a Documentum repository and that content being crawled and indexed by xPlore.

2.2 Planning for high availability

In a high-availability (HA) deployment, if an xPlore instance fails, then the secondary instance can replace it; CPS, indexing, and search operations continue on the replacement instance. An HA deployment provides the following benefits:

- Increased query availability because support is provided for failover.
- Redundancy if a host fails.

If you require xPlore high availability, you will typically also require high availability for your Documentum Server. Therefore, you should coordinate your Documentum Server and xPlore high availability deployments. Although many Documentum Server and xPlore high availability configurations are possible, a single repository served by two Documentum Servers is typical and can be used as a building block for more complex deployments. See the *OpenText Platform and Platform Extension Installation Guide* for more information about the different high-availability deployments and installation instructions for Documentum Server.

xPlore supports these kinds of high availability deployments:

- Spare instance - A spare instance is a warm instance that can be manually brought online when a running node fails. When the spare instance is started, it takes over the previous instance's transaction log and data, which must be accessible in a shared directory (the data and index directories must also be accessible to the spare instance). This functionality is included with xPlore.
- Active-Passive - Active-Passive is a failover technology that enables an application and its data to automatically start up on alternative (passive) hardware after the primary application fails. Microsoft, Redhat, and Veritas cluster servers are examples of active-passive clusters. Although failover is automated, active-passive cluster requires a heavy investment in duplicate servers but not duplicate disk storage.
- Active-Active - Active-Active HA consists of two or more complete, running xPlore deployments. Two or more different index agents send the same content to each deployment. Queries are directed to one of the active deployments or load balanced across both of them. In Active-Active HA, you must manage two or more deployments. This is the most expensive option because it requires duplicate servers and disk storage.

You can also deploy a hybrid model: Active-Passive or Active-Active clusters for the indexes only.

High availability and disaster recovery cost

The cost of high availability and disaster recovery is determined as follows:

- Number of servers

- Amount of disk storage:
 - Storage for backups. For example, you could have 3 days of backups that require 6 times the amounts of database storage. Cost can be lowered if backups are not needed for the indexes: you could refeed, which would increase the RTO.
 - Duplicate disk storage and I/O for active-active high availability. Active-passive usually uses duplicate servers but shared disk storage
- Cost of automation: Higher with an automatic failover system like Microsoft Failover Clustering than with manual failover (xPlore spare instance).

The following table illustrates how the cost increases as recovery time increases and automation becomes available.

Table 2-1: HA Strategy Comparison

HA Strategy	Cost	Recovery Time	Failover Automation	Notes
Spare instance	\$	Slowest	None	Not as automated but very inexpensive.
Active-Passive	\$\$	Medium	Yes	Highly automated; standard and well-known technology.
Active-Active	\$\$\$	Fastest	Yes	Has been in place the longest time. Requires more time to administer the duplicate system.

2.3 Two primary instances HA (active-active) process

The following diagram illustrates multi-instance HA using two xPlore primary instances on separate hosts.

 **Note:** The active-active HA configuration maintains indexing in case of a failover. You must manually restart query services in the standby Documentum Server.

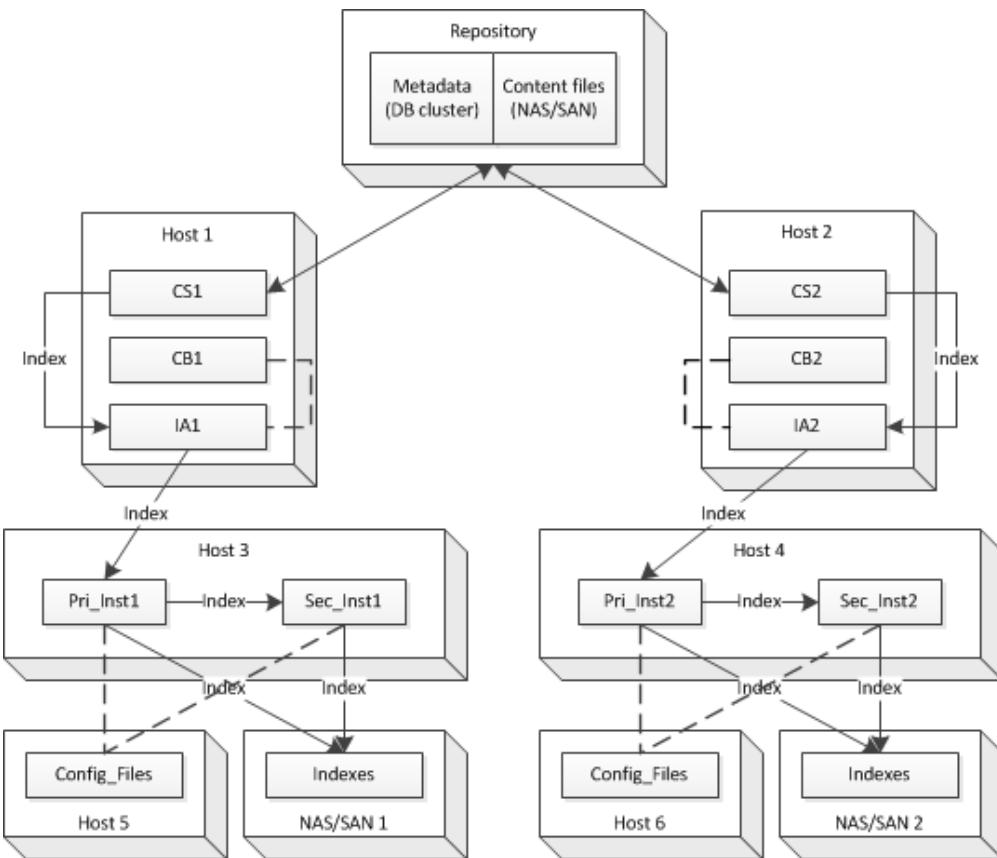


Figure 2-1: Dual xPlore primary instances HA before failover

- Host1: Primary Documentum Server host
 - Documentum Server (CS1)
 - Connection broker (CB1)
 - Index agent1 (IA1)
- Host2: Secondary Documentum Server host

- Documentum Server (CS2)
- Connection broker (CB2)
- Index agent2 (IA2)
- Host3: First primary xPlore host
 - Index agent (IA1) that connects to CS1
 - Primary instance1 (Pri_Inst1)
 - Secondary instance1 (Sec_Inst1)
- Host4: Second primary xPlore host
 - Index agent (IA2) that connects to CS2
 - Primary instance2 (Pri_Inst2)
 - Secondary instance2 (Sec_Inst2)
- Host5: First xPlore primary instance data, log, and configuration files (**Config_Files**), which must be writable from both the running instances as well as the spare instances.
- NAS/SAN: Full-text indexes for the first primary xPlore host
- Host6: Second xPlore primary instance data, log, and configuration files (**Config_Files**), which must be writable from both the running instances as well as the spare instances.
- NAS/SAN2: Full-text indexes for the second primary xPlore host

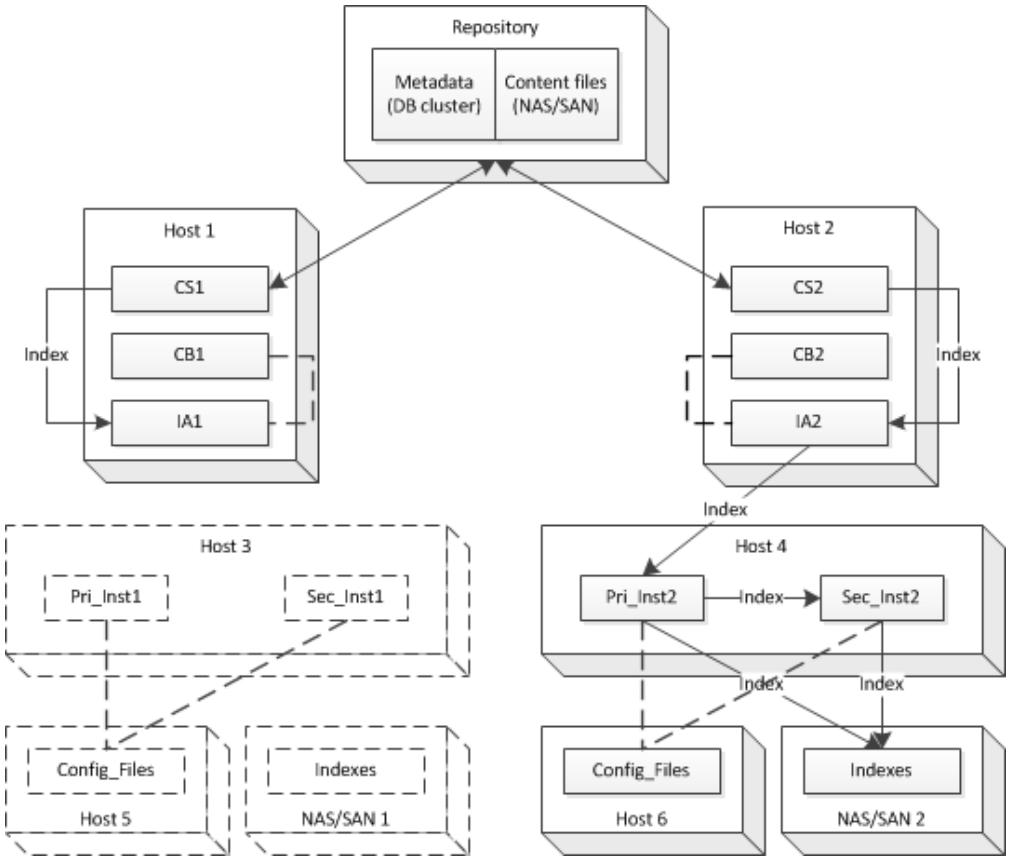


Figure 2-2: Two xPlore primary instances HA after failover

2.4 Planning for disaster recovery

In a multi-site disaster recovery (DR) configuration, the primary installation—the set of machines, installed software, and data that comprise the system—is replicated to a remote site. The remote site must be located far enough away from the primary installation so that the same catastrophe will not damage it as well.

Furthermore, when the remote site is brought online, it must be configured like the primary site—The remote instance must mirror the primary instance with the same instance name, port number, directory structure, and username/password combination.

The remote site does not have to recover to the exact same state as the primary site. For example, when a series of disk blocks are written on the primary site, they are sent to the remote site, but some of the blocks might be received out of order, some intervening blocks might be not be received, or the disk could be corrupt at the remote site. In order to bring the remote system online, all the received blocks must be contiguous and in the same order as they were written on the primary site. That is, bringing the remote system online is equivalent to auto-recovering xPlore at the

primary site. Although the remote site can run on reduced capacity compared to the primary site, OpenText strongly recommends, as a best practice, that the remote site run on hardware and software that has the same capacity as the primary one.

For example, EMC Recoverpoint is a multi-site disaster recovery product.

To plan for disaster recovery, OpenText strongly recommends performing regular backups, because bringing the remote system online is not guaranteed to succeed all the time (for example, the remote disk might have failed or data replication might have been unknowingly misconfigured). However, if no backup exists, the content can be re-fed from the original source system (for example, the Documentum repository).

2.5 Disaster recovery in a Documentum environment

When your full-text index installation fails, you recover it by restoring it to a previously consistent state (that is, a particular point in time) from your backups. Restoring to a particular point in time is also known as a point-in-time recovery.

Documentum Server and indexing backup coordination

If the associated Documentum Server has also failed, then you must coordinate the full-text index recovery with the Documentum Server recovery.

Backup of content and metadata in the Documentum repository must be coordinated for object consistency. Coordinated backups have the following characteristics:

1. Documentum Server is usually deployed in a disaster recovery configuration:
All data stores for the database, content, and full-text indexes are replicated from the primary site to a disaster recovery site. The disaster recovery site is backed up using a BCV-like (snapshot) technology.
2. Data replication from the primary site to the disaster recovery site completes within the RPO. The amount of data being replicated differs between the database, content, and full-text indexes, so the overall amount of time it takes for each one to be completely replicated also differs. Content generally takes the longest to replicate, and the database takes the shortest to replicate.
Consequently, the replication time for content is the maximum replication time for the entire deployment. The data replication rate depends on the network bandwidth and the oncoming load as well as a host of other environment factors..To set the RPO correctly, determine the maximum amount of time it takes for data (new and modified) to be completely replicated from the primary site to the disaster recovery site.
3. The full-text index backup interval is at least two times that of the database backup interval. In addition, you must wait for at least two times the entire deployment's maximum replication time before backing up the full-text index.



Note: Since the full-text indexes can be restored point-in-time from the database and content, the full-text indexes could be backed up as

infrequently as once a week whereas the database and content could be backed up many times a day. Full-text index backups must be performed frequently enough to meet the RTO. That is, the full-text index point-in-time recovery is determined by the amount of data received during the full-text index backup interval.

 **Note:** Full-text indexes can be re-indexed but could take weeks for a really large index.

In the following example, the recovery point objective is 8 hours. After replication of the database (content and metadata) and full-text indexes (hours 0–2), a point of failure occurs at 9.5 hours. The database is restored to the recent backup (0.5 hours before failure). The content is restored to the recent backup (7.5 hours before failure). The index is restored to 7.75 hours before failure. The Documentum Server administrator handles the content synchronization between database and metadata. For xPlore, you must refeed all content that was indexed since the backup was completed. Run *ftintegrity* using a start and stop window to get the object IDs.

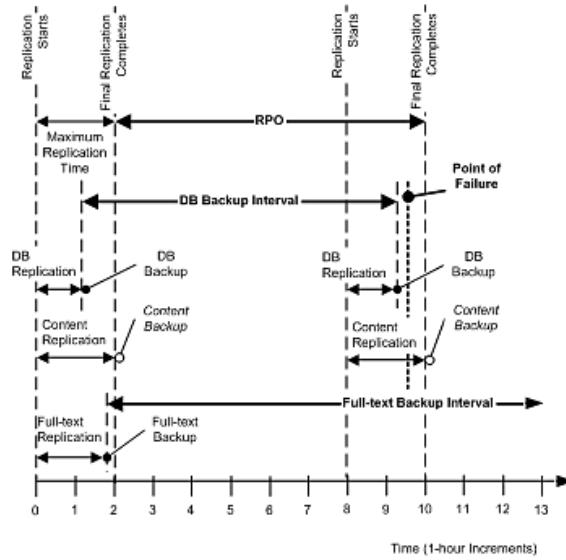


Figure 2-3: RPO for disaster recovery site



Notes

- Italics: The operation is optional (Content Backup).
- The time scale and line lengths of the operations indicate only a rough order of magnitude—that is, they are not typical nor do they represent a recommendation.

Restoring the xPlore indexes

See the chapter Backup and Restore in *OpenText Documentum xPlore Administration and Development Guide*.

Chapter 3

Installing xPlore

3.1 Before you install

- Install xPlore in a test environment first. Performance-test your xPlore installation with expected customer workloads, queries, and document profiles before moving your installation to a production environment.
- In Linux, enable the random generator by starting it as root:

```
/sbin/rngd -b -r /dev/urandom -o /dev/random
```
- Documentum Server: Configure at least one Documentum repository and one connection broker.
- Windows 2008 Server: For support of xDB command line utilities, install the Microsoft Visual C++ redistribution package.
- Linux: Installation of xPlore in console mode is not supported. Set up your environment to support GUI installation. Installation on 64-bit Linux requires the following RPM packages before xPlore installation:
 - libXp-<version>-<release>. <architecture> (e.g. libXp-1.0.0-15.1.el6.i686)
 - libXi-<version>-<release>. <architecture> (e.g. libXi-1.3-3.el6.i686)
 - libXtst-<version>-<release>. <architecture> (e.g. libXtst-1.0.99.2-3.el6.i686)
 - libXt-<version>-<release>. <architecture> (e.g. libXext-1.1-3.el6.i686)

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

- All instances must be installed on the same operating system (Windows or Linux).
- To avoid permission-related errors, the same installation owner (same username, same password) must install all xPlore instances, index agents, and CPS instances. On Windows, the installation owner must also be in the same Windows domain.
- Install a primary instance before installing secondary instances. Secondary instances are optional.
- Download and install the supported Oracle JDK 11 or OpenJDK 11 version from Oracle website. OpenJDK documentation contains detailed information.

See the *Documentum Server* chapter in *OpenText Documentum Platform and Platform Extensions Installation guide* for details about using Anonymous SSL communication.

- Documentum xPlore can be installed on computers that run the following operating system code pages:
 - For U.S. and Western European sites, ISO-8859_1 (Latin-1)
 - For Korean sites, EUC-KR For Japanese sites that use Linux, EUC-JP
 - For Japanese sites that use Windows, Shift_JIS
 - For Chinese sites with locale zh, ms936
 - For Russian with locale ru, Windows-1251



Note: Ensure that xPlore is configured for the English locale first if the Documentum Server locale is configured for other locales.

The *OpenText Documentum Platform and Platform Extensions Installation guide* contains more information about configuring internationalization settings on Documentum Server.

3.2 Installing the xPlore primary instance with Documentum Server 7.3 and later

To install the xPlore primary instance with Documentum Server 7.3 and later:

1. Unzip one of the following files to a temporary directory depending on your platform:
 - Documentum_xPlore_<version>_windows_x64.zip
 - Documentum_xPlore_<version>_linux_x64.zip
2. Run `setup.exe` (Windows) or `setup.bin` (Linux) to launch Installer and follow online instructions to complete the installation process.
 - a. **Where would you like to install?**: Install xPlore and its components to a root directory and choose a path that does not contain spaces. The default installation directory is `C:\xPlore` on Windows and `<$HOME>/xPlore` on Linux.

On Windows, use a local path during installation. For a multi-instance configuration, this drive must be accessible to other xPlore instances by UNC path. For example, you install to local path `F:\xPlore`. When you configure the primary instance, you specify the xPlore data and config paths as `\<hostname>\F$\xPlore\data` and `\<hostname>\F$\xPlore\config` respectively.
 - b. **Watchdog Administrator Information**: Enter an SMTP server and email address. The SMTP server cannot be a Microsoft Exchange server.

This step is optional. You can click Next to proceed and safely ignore the error prompt. You can set the SMTP server and email address at a later time by setting the properties of the `SendMailTask` task in the file `dsearch-watchdog-config.xml` located in `xplore_home/watchdog/config`.

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. Create the primary xPlore instance.

Run `xplore_home\setup\dsearch\configDsearch.bat` (Windows) or `xplore_home/setup/dsearch/configDsearch.sh` (Linux) and follow the online instructions to complete the installation process.

- a. **Select Configuration Mode:** Choose **Create Primary Server Instance**.
- b. **(Windows only) 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. Enter the fully qualified domain name of the xPlore primary instance host.
- d. **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 Database User** is also the xDB Administrator password.



Note: The database password can be up to 31 characters long and must follow these rules:

- Cannot begin with # (Sharp)
- Cannot contain the following special characters:
 - Windows: > < % | ^ & () ' "
 - Linux: < > \$ ` ! & * () | ; ' '

- e. **Default Admin User Information:** Enter a username and password. Use these credentials to log in to the xPlore Administrator console as superuser. The password length and rules of the database password also apply to this administrator password.
- f. **Dsearch Data and Config 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. For NAS-based storage, you can map the same storage to the same path for all hosts.



Note: Instability network for NAS-based storage might corrupt xPlore data. Ensure you test it thoroughly before moving it to the production.

Windows: If you want to deploy multiple xPlore instances or there might be a future need to do so, you must specify UNC paths accessible to other xPlore instances; for example, `\ \<hostname>\D$\xPlore\data` and `\ \<hostname>\D$\xPlore\config` (when `<xplore_home>` is `D:\xPlore`).



Note: On Windows, it is highly recommended that you use UNC paths for these directories even if you are not deploying multiple xPlore instances at the moment.

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

Configuration Directory stores configuration information. Default: *xplore_home*/config.

4. Start the primary xPlore instance:

- In *xplore_home/*<wildfly_version>/server/, run the script *startPrimaryDsearch.cmd* (Windows) or *startPrimaryDsearch.sh* (Linux).
- On Windows, you can also start the service **Documentum xPlore PrimaryDsearch**.



Note: On Windows, if you start an xPlore instance using the command, the status of the xPlore service may not be accurately reflected in the Windows Management Console.

5. For each Documentum repository, install and configure an xPlore index agent. See “[Configuring the index agent](#)” on page 45.
6. 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).

For example:

```
http://server.opentext.com:9300/dsearch
```

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

```
The xPlore instance PrimaryDsearch [version=current_version] normal
```

After installing the primary instance, you can optionally proceed to install secondary and spare instances. See “[Installing a secondary or spare instance](#)” on page 33.

3.3 Validating installation

3.3.1 Starting xPlore Administrator

The primary xPlore instance includes the xPlore administrator web application. Start all xPlore instances in order to see their status in xPlore administrator.

1. Open your web browser and enter the following. Substitute the hostname and port for the primary instance.

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

2. Again enter the username and password for the administrator.

3.3.2 Testing indexing

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

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

1. In xPlore Administrator, select **Diagnostic and Utilities > Upload Testing Document**.
2. Specify the appropriate values in these fields:
 - **Domain**
 - **Collection** - the collection to which to add the document.
3. Select one of these options:
 - **Option 1:** Upload a document when the xPlore Administrator and the indexing service are on the same host. Specify 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:** (Optional) 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.
 - **Option 2:** Upload a document from a shared data store or using a remote CPS instance. Specify the URL, using the file protocol syntax, to the document in the **Remote File** field. All other fields are identical to **Option 1**.
 - **Option 3:** Click **Specify raw XML**. For example, enter some XML in the DFTXML format.

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) and *<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 “Testing search” on page 30.

3.3.3 Testing search

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

1. In xPlore Administrator, select **Diagnostic and Utilities > Test Search**.
2. Choose one:
 - **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. Verify that the correct results are displayed.
 - If you use a Webtop client to test search, click Control-Edit to see the XQuery that was used in the search.
 - 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 'dmadmin'  
ENABLE(FTDQL_DATA)
```

If the Documentum 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') ) )]
```

3.4 Sharing xPlore directories from the primary instance host

In a multi-instance configuration, the following directories on the primary instance must be accessible to all secondary instances.

- xPlore data directory, default: `<xPlore_home>/data`
- xPlore configuration directory, default: `<xPlore_home>/config`
- Index agent temporary storage location, default: `<xPlore_home>/<wildfly_version>/server/DctmServer_Indexagent/data/Indexagent/export/IndexAgent`

This is not a problem if the primary and secondary instances are installed on the same host, but if they are installed on separate hosts, you must share the these directories from the primary instance host so that they can be accessed by secondary instances from other hosts.

On Windows, these directories must be shared on the network and can be accessed with read and write permissions through a UNC path from the hosts where secondary instances will be installed. If you specified local paths instead of UNC paths for xPlore data and configuration directories when you installed the primary instance, you must perform the following additional steps to share these directories through symbolic links without having to reinstall the primary instance:

1. On the secondary instance host, enable the Remote to Local and Local to Remote symbolic link evaluations by executing the following commands in the command prompt window:

```
fsutil behavior set SymlinkEvaluation R2L:1 L2R:1
```

Local-to-remote is a computer accessing a local symbolic link that points to a Universal Naming Convention (UNC) path using the server message block (SMB) protocol. Remote-to-local is a computer accessing a remote symbolic link that points to a local file or folder using SMB.

2. Manually create directories in which you want to create symbolic links if they do not exist yet; for example:
 - Create `C:\xPlore` so that config and data symbolic links can be created in it
 - Create `C:\xPlore\<wildfly_version>\server\DtcmServer_Indexagent\data\Indexagent\export` so that the Indexagent symbolic link can be created in it
3. In the command prompt window, execute commands to create symbolic links to remote shared UNC paths; for example:

```
mklink /D C:\xPlore\config \\primary_host\C$\xPlore\config  
mklink /D C:\xPlore\data \\primary_host\C$\xPlore\data  
mklink /D C:\xPlore\dblog \\primary_host\C$\xPlore\dblog
```

```
mklink /D C:\xPlore\wildfly17.0.1\server\DtcmServer_Indexagent\data\Indexagent\export\IndexAgent \\primary_host\C$\xPlore\wildfly17.0.1\server\DtcmServer_Indexagent\data\Indexagent\export\IndexAgent
```

4. The symbolic links are created and appear as directory shortcuts. You can then install the secondary instance and configure it to use these links to transparently access the shared directories on the primary instance host as if accessing the remote UNC paths directly.

On Linux, share the directories from the primary instance host to secondary instance hosts using the Network File System (NFS) service. The following steps assume that *<xplore_home>* is */root/xPlore* on both the primary instance host (host1) and secondary instance host (host2).

1. Share the NFS system on the primary instance host.

- a. Add the directories to export to the */etc/exports* file; for example:

```
/root/xPlore/config *(sync,rw,no_root_squash)
/root/xPlore/data *(sync,rw,no_root_squash)
/root/xPlore/dblog *(sync,rw,no_root_squash)
/root/xPlore/wildfly17.0.1/server/DtcmServer_Indexagent/data/Indexagent/export/
IndexAgent *(sync,rw,no_root_squash)
```

- b. Export the shared file system using the following command:

```
# /usr/sbin/exportfs -a -v
```

- c. Turn on the NFS service using the following commands:

```
# /etc/init.d/nfs start
```

```
# /etc/init.d/nfslock start
```

The NFS service should now be running and ready to share the directories with other computers on your network.

2. Mount the NFS system on the secondary instance host.

- a. Create the following mount points if the directories do not exist:

```
# mkdir /root/xPlore/config# mkdir /root/xPlore/data# mkdir
/root/xPlore/dblog# mkdir /root/xPlore/<wildfly_version>/
server/DtcmServer_Indexagent/data/Indexagent/export/IndexAgent
```

- b. Manually mount the directories shared from the primary instance host using the following commands:

```
# mount -t nfs host1:/root/xPlore/config /root/xPlore/config#
mount -t nfs host1:/root/xPlore/data /root/xPlore/data# mount -t
nfs host1:/root/xPlore/dblog /root/xPlore/dblog# mount -t nfs
host1:/root/xPlore/<wildfly_version>/server/DtcmServer_
Indexagent/data/Indexagent/export/IndexAgent /root/xPlore/
<wildfly_version>/server/DtcmServer_Indexagent/data/
Indexagent/export/IndexAgent
```



Note: The mount is temporary and is not remounted when you reboot your computer.

- c. To make the mount permanent and automatic each time you start your Linux system, add the following entries to the /etc/fstab file:

```
host1:/root/xPlore/config /root/xPlore/config nfs auto 0 0
host1:/root/xPlore/data /root/xPlore/data nfs auto 0 0
host1:/root/xPlore/dblog /root/xPlore/dblog nfs auto 0 0
host1:/root/xPlore/<wildfly_version>/server/DctmServer_Indexagent/data/Indexagent/export/IndexAgent /root/xPlore/<wildfly_version>/server/DctmServer_Indexagent/data/Indexagent/export/IndexAgent nfs auto 0 0
```

- d. The directories of the xPlore primary instance are now accessible to the secondary instance.

3.5 Installing a secondary or spare instance

For all xPlore instances, index agents and CPS instances:

- Install a primary instance before installing secondary instances. Secondary instances are optional.
- To avoid permission-related errors, the same installation owner (same username, same password, and same Windows domain) must install all instances.
- All instances are on the same operating system (Windows or Linux).
- On Windows hosts, use universal naming convention (UNC) to specify shared folders.
- Make sure you can access the following directories on the primary instance host from the secondary instance host:
 - xPlore data directory, default: <xplore_home>/data
 - xPlore configuration directory, default: <xplore_home>/config
 - Index agent temporary storage location, default: <xplore_home>/wildfly_version/server/DctmServer_Indexagent/data/Indexagent/export/IndexAgent

If these directories are not accessible from the secondary instance host, share them out first. See “[Sharing xPlore directories from the primary instance host](#)” on page 31.

You will be able to create additional storage for each instance, either before you install secondary instances or after. You can bind an instance to the new storage location. Storage locations must be accessible to all instance. Use xPlore Administrator to create a storage location and bind an instance to it. See *OpenText Documentum xPlore Administration and Development Guide*.

The following instructions assume that you are installing the primary and secondary instances on different hosts. If you are installing a secondary instance on the same host as the primary instance, skip to step 4.

1. Make sure you have mapped each hostname to its IP address in the hosts file. The hosts file is <%SystemRoot%>\Windows\system32\drivers\etc\hosts on Windows and /etc/hosts on Linux. Use the following example to create mappings in the hosts file:

```
10.32.112.233 Primary  
10.32.112.235 Node2
```

2. Unzip one of the following files to a temporary directory:
 - Documentum_xPlore_<version>_linux_x64.zip
 - Documentum_xPlore_<version>_windows_x64.zip
3. Run setup.exe (Windows) or setup.bin (Linux) to launch Installer and follow online instructions to complete the installation process.
 - a. **Where would you like to install?**: Install xPlore and its components to a root directory and choose a path that does not contain spaces. The default installation directory is C:\xPlore on Windows and <\$HOME>/xPlore on Linux.
 - b. **Get Watchdog Administrator Information**: Click Next to proceed and safely ignore the error prompt.
4. Start the primary instance on the primary host:
 - Windows: The **Documentum xPlore <PrimaryDsearch> (Primary)** Windows service, or xplore_home\<wildfly_version>\server\startPrimaryDsearch.cmd
 - Linux: xplore_home/\<wildfly_version>/server/startPrimaryDsearch.sh
5. Run xplore_home\setup\dsearch\configDsearch.bat (Windows) or xplore_home/setup/dsearch/configDsearch.sh (Linux) and follow the online instructions to complete the installation.
 - a. **Select Configuration Mode**: Choose **Create Secondary Server Instance**
 - b. If you are installing a secondary instance to serve as a 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.

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

 - c. (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.

- d. Enter the *fully qualified domain name* of the xPlore secondary instance host.



Note: The name of a secondary instance must begin with an alphabetic character followed by any combination of alphanumeric characters. If the name does not conform to this requirement, for example, the name contains the underscore (_) character, the system returns the “failed to register *instance_name*” error.

- e. **Primary Host Information:** **Primary Host** is the DNS name of the primary instance host, and **Primary Port** default is 9300.
- f. **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 (e.g. 9500). The next 100 consecutive ports must be available. For **Password for Admin User**, use the xPlore administrator password.

In **Select Roles**, specify which roles the secondary instance will serve.

- g. **Dsearch Log Directory (Secondary Node):** In the **Transaction Log Directory** field, specify a directory for transaction log files. If you install multiple xPlore instances on separate hosts, the path must be accessible to all instances and conform to UNC on Windows. If you specify a UNC path, transaction log files will be stored in <UNCPath>\<instance_name>_log; if you do not specify a directory, the transaction directory<instance_name>_log will be under the configuration directory by default.

6. Restart all instances including the primary instance. To start a secondary instance, use one of the following methods:

- Run the script *startSecondary_instance.cmd* (Windows) or *startSecondary_instance.sh* (Linux) in *xplore_home/<wildfly_version>/server*.
- On Windows, you can use the script or start the service **Documentum xPlore <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.

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 xPlore instance DsearchNode2 [version=current_version] normal
```

3.6 Configuring SSL connections

xPlore supports SSL connections to xPlore Administrator, index agent, and between index server and Documentum Server (7.1 and later). You can configure SSL in several ways:

- “Configuring SSL connections using script (recommended)” on page 36
- “Configuring SSL for remote CPS” on page 39

3.6.1 Configuring SSL connections using script (recommended)

You can easily enable/disable SSL for secure connections to index server and/or index agent using the SSL configuration script <xplore_home>/dsearch/admin/scripts/ConfigSSL.groovy. Follow these steps to perform the configuration:

1. Shut down all the index server and index agent instances.
2. Run the SSL configuration script using the following command (xplore.bat on Windows and xplore.sh on Linux):

```
xplore.bat -f scripts/ConfigSSL.groovy [-h | -help] [-enable | -disable | -dual] [-component [ALL | IS | IA]][-selfsigned] [-alias <alias>] [-dnname "CN=<CN>,OU=<OU>,O=<O>,L=<L>,ST=<ST>,C=<C>"] [-keystore "file/my.keystore"] [-storepass <storepass>] [-indexserverconfig <config_file_path>] [-isname <indexserver_name>] [-ianame <indexagent_name>] [-iaport <indexagent_port>]
```

For example:

- To enable HTTPS for both index server and index agent using a self-signed certificate:

```
xplore.bat -f scripts/ConfigSSL.groovy -component ALL  
-selfsigned -dnname CN=xplore.abc.com,OU=ABC,O=ABC,L=SH,S=SH,C=CN
```

- To disable HTTPS for both index server and index agent:

```
xplore.bat -f scripts/ConfigSSL.groovy -disable -component ALL
```



Note: You must execute the SSL configuration script on the host where the instances to configure is installed. For example, suppose the primary index server and index agent instances run on host A, a secondary index server instance runs on host B, and a secondary index agent instance runs on host C. To configure SSL for all these instances, you must execute the script on all the three hosts.

The SSL configuration options are as follows:

Option	Description
-h -help	Display the script help message
-enable	Enable HTTPS and disable HTTP. This is the default option.
-disable	Disable HTTPS and enable HTTP
-dual	Enable HTTPS but HTTP is not disabled
-component [ALL IS IA]	<p>Specify whether to perform the configuration for index server (IS), index agent (IA), or both (ALL); Default: IS</p> <p>If you set this option to IS or ALL, all the index server instances on the host will be configured unless you specify a single index server instance using the <i>-isname</i> option.</p>
-selfsigned	Enable SSL using a self-signed certificate. A private key will be generated.
-alias <alias>	Specify a unique key alias for the instance to configure to prevent overwriting existing keys. Default: "xplore" or "ia" (when option <i>-component IA</i> is specified)
-dname <distinguished_name>	Distinguished name for the certificate. If you do not specify this option when enabling SSL, you will be prompted to provide this information.
-keystore "file/my.keystore"	Specify a keystore that stores all the required certificates for SSL encryption. The certificates must be imported into the JDK keystore cacerts (<xplore_home>/JAVA_LINK/lib/security/cacerts). Only jks format keystore is supported.
-storepass <storepass>	If you use the <i>-keystore</i> option, specify a keystore password with a minimum length of six characters; default: changeit
-indexserverconfig <config_file_path>	The full path to indexserverconfig.xml; default: <xplore_home>/config/indexserverconfig.xml
-isname <indexserver_name>	If you want to configure SSL connections for just one index server instance, specify the instance name using this option; otherwise, all index server instances are configured with their instance name and port information retrieved from indexserverconfig.xml.

Option	Description
<code>-ianame <indexagent_name></code>	<p>Use this option for both index server and index agent configuration.</p> <p>For index agent configuration, specify the name of the index agent instance to configure.</p> <p>For index server configuration, specify the appropriate index agent instance name on the local host so that the script can update the values for <code>dsearch_qrserver_protocol</code> and <code>dsearch_config_port</code> in <code>indexagent.xml</code>.</p> <p>However, if the index agent instance resides on a remote host, you must modify the <code>indexagent.xml</code> file manually.</p> <p>If not specified, the default name Indexagent will be used for both index agent and index server configurations.</p>
<code>-iport <indexagent_port></code>	<p>The port currently used by the index agent you want to configure; default: 9200 (HTTP), 9202 (HTTPS)</p> <p>If you enable SSL, the HTTPS port number will be <code><indexagent_port>+2</code>.</p> <p>If you disable SSL, the HTTP port number will be <code><indexagent_port>-2</code>.</p>

3. If the index agent is not on the same host with the primary index server, edit `indexagent.xml` in `<xplore_home>/<wildfly_version>/server/DctmServer_Indexagent/deployments/IndexAgent.war/WEB-INF/classes`:

```

<parameter>
  <parameter_name>dsearch_qrserver_protocol</parameter_name>
  <parameter_value>HTTPS</parameter_value>
</parameter>
<parameter>
  <parameter_name>dsearch_config_host</parameter_name>
  <parameter_value>myserverhost.domainname.com</parameter_value>
</parameter>
<parameter>
  <parameter_name>dsearch_config_port</parameter_name>
  <parameter_value>9302</parameter_value>
</parameter>

```

4. For index server, change the `dm_ftengine_config` object in the Documentum Server to update the values of `dsearch_qrserver_protocol`, `dsearch_qrserver_port`, and `dsearch_config_port` parameters.

- a. Get the `dm_ftengine_config` object:

```
API>retrieve,c,dm_ftengine_config
```

- b. Find the index value of `dsearch_qrserver_protocol` (for example, 6).

```
API>dump,c,1
```

- c. Set the values for dsearch_qrserver_protocol, dsearch_qrserver_port, and dsearch_config_port. This example uses the index value returned by the dump command:

```
API>set,c,1,param_value[6]
SET>HTTPS
```

Example to set the SSL ports for dsearch_qrserver_port:

```
API>set,c,1,param_value[0]
SET>9302
```

Example to set the SSL ports for dsearch_config_port:

```
API>set,c,1,param_value[1]
SET>9302
```



Note: Check the index values before you set values.

- d. Save the changes made to the dm_ftengine_config object:

```
API>save,c,1
```

5. Restart the Docbase, index server, and index agent instances.

3.6.2 Configuring SSL for remote CPS

The SSL configuration script does not configure SSL on remote CPS instances. To configure SSL for a remote CPS, you must complete the following steps manually:

1. Create a certificate keystore named `my.keystore` using the Java `keytool` utility.

- On Linux:

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

- On Windows:

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

When the `keytool` utility asks for a Common Name (CN) value (first and last name), enter the fully qualified name of the application server host as referenced by your browser or web service consumer. Use `changeit` as the password and `tomcat` as the alias. The keystore named `my.keystore` is created and stored in the user home directory by default (the directory specified by the `user.home` system property).

2. Export the certificate with the following command:

```
%JAVA_HOME%\bin\keytool.exe -export -alias xplore
-keystore my.keystore -file mycerts.cer -storepass changeit
```

The command output displays that the certificate was stored in the given filename.

3. Import the certificate with the following command:

```
%JAVA_HOME%\bin\keytool.exe -import -trustcacerts
-alias xplore -keystore <jdkCACERTSpath> -file mycerts.cer
-storepass changeit
```

Where <jdkCACERTSpath> is:

```
<xplore_home>/JAVA_LINK/lib/security/cacerts
```

4. Copy the file my.keystore to:

```
<xplore_home>/<wildfly_version>/server/
DctmServer_<><remote_cps_instance_name>>/configuration
```

For example:

```
<xplore_home>/<wildfly_version>/server/DctmServer_RemoteCPS/
configuration
```

5. Enable an HTTPS port for the xPlore instance in WildFly:

a. Edit the file standalone.xml that is in the folder:

```
<xplore_home>/<wildfly_version>/server/
DctmServer_<><remote_cps_instance_name>>/configuration
```

b. Configure the new keystore as a server identity for SSL in the security-realms section of standalone.xml.

```
<management>
  <security-realms>
  ...
    <security-realm name="sslRealm">
      <server-identities>
        <ssl>
          <keystore path="C:\xPlore\<wildfly_version>\server\
DctmServer_RemoteCPS\configuration\my.keystore"
            keystore-password="password" alias="xplore"/>
        </ssl>
      </server-identities>
    </security-realm>
  </security-realms>
</management>
```

c. Add the https-listener to the undertow subsystem and change the http listener name to http-listener:

```
<subsystem xmlns="urn:jboss:domain:undertow:2.0">
  <server name="default-server">
    <http-listener name="http-listener" socket-binding="http"
      redirect-socket="https"/>
    <https-listener name="default" socket-binding="https"
      security-realm="sslRealm"
      enabled-cipher-suites="TLS_RSA_WITH_AES_128_GCM_SHA256"/>
  ...
  </server>
...
</subsystem>
```

6. Change the port for the HA heartbeat on Windows:

a. Edit the JBoss service configuration file which is in the folder:

```
<xplore_home>/<wildfly_version>/server/serviceConfig/
<Instance_name>/conf
```

For example:

```
C:/xPlore/wildfly17.0.1/server/serviceConfig/RemoteCPS/
conf/DmRemoteCPS.xml
```

- b. Change the port number of the heartbeat URL from the HTTP port number to the HTTPS port number. For example:

```
<heartbeat>
  <url>https://CPSMachine:9902</url>
</heartbeat>
```

7. Delete the WildFly cache folder to clear the cache:

```
<xplore_home>/<wildfly_version>/server/
DctmServer_<remote_cps_instance_name>/tmp/work
```

8. Start the remote CPS.
9. Verify if SSL is working by accessing https://<hostname>:<httpsport>/cps/ContentProcessingService?wsdl. For example: https://localhost:9902/cps/ContentProcessingService?wsdl.
10. Add the secure URL of the remote CPS in the Admin console and restart the xPlore nodes.



Note: Add the same URL that you used in Step [on](#).

11. Test the index and search functions using the Admin console.

3.7 Adding CPS Instances

By default, a CPS instance is installed when you install a primary or secondary instance. Requests are distributed round-robin to each CPS instance. You can add CPS instances to improve performance and scalability. For instructions on adding a CPS instance that is dedicated to processing requests for an individual xPlore instance, see “[Configuring CPS dedicated to indexing or search](#)” on page 43.

For all xPlore instances, index agents and CPS instances:

- Install a CPS instance on the same operating system as all other xPlore instances.
 - Install a primary xPlore instance before installing a secondary CPS instance.
 - To avoid permission-related errors, the same installation owner (same username, same password, and same Windows domain) must install all instances.
 - On Windows, use universal naming convention (UNC) to specify shared folders (do not use mapped drives).
1. If you are installing a CPS instance on a machine on which you have not installed xPlore, run xPlore Installer first; otherwise, skip this step.
 - a. Unzip one of the following files to a temporary directory:
 - Documentum_xPlore_<version>_linux_x64.zip

- Documentum_xPlore_<version>_windows_x64.zip
- b. Run `setup.exe` (Windows) or `setup.bin` (Linux) to launch Installer and follow online instructions to complete the installation process.
2. Run `<xplore_home>\setup\dsearch\configDsearch.bat` (Windows) or `<xplore_home>/setup/dsearch/configDsearch.sh` (Linux).
3. **Select Configuration Mode:** Choose **Create Content Processing Service only**.
4. Enter the **installation owner password**.
5. Specify the **fully qualified domain name** of the xPlore Dsearch host, where the xPlore instance you are creating CPS for is installed. The default is the current host name. If you are creating CPS for the primary xPlore instance, specify the primary xPlore instance host name.
6. **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; for example, change the port to 9500. The next 100 consecutive ports must be available. Use the same password as the xPlore administrator.
7. Proceed to complete the configuration.
8. Verify that the `<export_file_path>` location in the remote CPS configuration. `.xml` file is accessible by all instances. This file is located in the CPS host directory `<xplore_home>/dsearch/cps/cps_daemon`. If you change this file, restart all instances including the remote CPS.
 - Windows: Use universal naming convention (UNC) to specify the path; for example: `file:///C$/xplore/dsearch/cps/cps_daemon/export`.
 - Linux: Mount the `<export_file_path>` directory shared from the remote CPS.
9. Start the CPS instance. `<CPS_instance>` is the value you specified for **Server Name** you specified for the CPS instance.
 - Windows:
 - The **Documentum xPlore <CPS_instance> (CPS)** Windows service, or
 - `<xplore_home>\<wildfly_version>\server\start<CPS_instance>.cmd`
 - Linux: `<xplore_home>\<wildfly_version>\server\start<CPS_instance>.sh`
10. 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/cps/ContentProcessingService?wsdl`
11. To test the remote CPS service using the WSDL testing page, open a browser and specify the following syntax:

```
http://hostname:port/cps/ContentProcessingService?wsdl
```

A page showing the CPS service configuration is displayed.

3.8 Configuring CPS dedicated to indexing or search

By default, every xPlore instance has a local CPS service. All CPS services receive processing requests on a round-robin basis. For high-volume environments with multiple xPlore instances, you can configure one or more CPS services to handle all processing requests for a specific xPlore instance.

You can configure additional CPS instances that are dedicated to indexing in xPlore administrator, for high ingestion requirements, or dedicated to search, for heavy search usage.

1. Expand **Services** in the left panel, click **Content Processing Service**.
2. Click **Add** on the Content Processing Service page.
3. Select **local or remote instance** in the **Add Service** window, and then select **node instance and usage** in the drop-down list. For remote instance, you must specify its URL. Remote CPS service is always configured for dedicated indexing or search for specific xPlore instances.

Make sure that you have a remaining global CPS instance. For example:

```
<content-processing-services context-characters="!.,;?'&quot;"  
special-characters="@#$%^~`*&gt;:()-+=&lt;&gt;/\[]{}">  
  <content-processing-service usage="all" url="local"/>  
</content-processing-services>
```

3.9 Deleting an xPlore or CPS instance



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

The primary instance and the instance to be deleted must be running.

1. On the machine on which the instance is installed, run `<xplore_home>\setup\dsearch\configdsearch.bat` (Windows) or `<xplore_home>/setup/dsearch/configdsearch.sh` (Linux) and follow the instructions. `<xplore_home>` is the xPlore root directory.
2. On the **Select Configuration Mode** page, select the **Delete Existing Server Instance** option.
3. On the **Delete Instance** page, in the **Instance** field, select the name of the instance to delete.
4. 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 Index Agents

4.1 Preparing index agent installation

Run the xPlore installer to install an index agent for each repository.

Index agent requirements

- To avoid permission-related errors, the same installation owner (same username, same password, and same Windows domain) must install all xPlore instances, index agents, and CPS instances.
- 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 xPlore primary instance with Documentum Server 7.3 and later](#)” on page 26.
- Verify that the xPlore primary instance, Documentum Server, and Documentum global registry are started. After successfully starting up Documentum Server and the connection broker, wait several minutes (typically 15 minutes) before configuring a Documentum index agent.
- Each Documentum Server requires its own index agent.

Specify an alternate location for index agent storage

You can create a domain and select its storage location before you configure the Documentum index agent. You can then select the alternate storage location during index agent configuration. The domain name must match that of the Documentum Server repository.

4.2 Configuring the index agent

Run the xPlore installer to install an index agent for each repository. Then configure index agents using the following instructions.



Note: SSL connections should be enabled only after creating the Index Agent(s). If the xPlore and Index Agent(s) are already running in SSL mode, you need to disable the SSL connections for xPlore and the Index Agent(s) first and then add a new Index Agent.

1. Run `<xplore_home>\setup\indexagent\configIndexagent.bat` (Windows) or `<xplore_home>/setup/indexagent/configIndexagent.sh` (Linux).
2. *Configuration Mode:*
 - To create an index agent for the primary xPlore deployment, choose **Create a New Indexagent**.

- To create an index agent for a standby xPlore deployment in an HA configuration, choose **Create a new indexagent to standby xPlore Server**.
3. (Windows only) **Installation Owner Password**: Specify the installation owner password.
 4. **Server Instance Information:**
 - **Server Name**: Unique name for this index agent.
 - **HTTP Port**: Port number (default: 9200) for HTTP requests. The installer validates that the next 20 consecutive ports are available. Index agents on the same host must have unique port numbers.
 - **Password for Admin User**: You can use the same password as the xPlore administrator's.
 5. **Fully Qualified Domain Name**: Enter the fully qualified domain name of the index agent host.
 6. **Primary Dsearch Server Information**: Specify the **Dsearch host** and **Dsearch port** (default: 9300) of the xPlore primary instance.
 7. **Enter the directory for local content area**: Specify a directory path to a directory in which content is temporarily stored during indexing.

The directory must be accessible to all xPlore instances and external CPS, and writable from the index agent host by the current installation owner.

On Windows, if the primary xPlore instance or any of its secondary instances is on a different host than the index agent, the directory must be a network location and conform to UNC.
 8. **Connection Broker Information**: Enter the host name and port for the Documentum Server connection broker.

If the Documentum Server uses non-anonymous or certificate-based SSL for secure connection, select **Use certificates** and specify the following DFC truststore information:

 - If you use a single JDK default truststore, select **Use Certificates**. Leave **TrustStore** and **Password** as is and select **Use Default Java TrustStore**.
 - If you use your own truststore, select the **Use Certificates** option and supply the **TrustStore** pathname and **Password**. Leave the **Use Default Java TrustStore** option unselected.
 9. When the Index Agent Configurator successfully connects to the connection broker, additional fields appear. Enter the following information:
 - **Select Docbase**: Choose a Documentum repository to which the index agent connects.
 - **User Name**: Enter the Documentum Server installation owner user name.
 - **Password**: Enter the Documentum Server installation owner password.

10. **Global Registry Information:** Select the global repository and enter login information.
11. If you are installing a standby index agent, you are presented with a screen in which to **set fulltext object configuration**.
Accept the default values for **fulltext user**, **fulltext index name**, and **fulltext engine name**.
If the default value have already been used, such as by FAST in a dual mode configuration, specify a unique value.
12. If you install the index agent in a multi-instance xPlore environment, restart all xPlore instances after installation is complete.
13. Restart the Documentum Server to enable the index agent to respond appropriately to a change in the Documentum Server's dormancy status.

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

4.3 Starting the index agent

Set up index agent filters before you start indexing or 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.

1. Start with a script or service. *IndexAgent* is the value you specified for **Server Name**.
 - *Linux*: Run *xplore_home\<wildfly_version>\server\startIndexAgent.sh*.
 - *Windows*: Start the index agent Windows service (or restart it, if it is already running) **Documentum <IndexAgent>**, or run *xplore_home\<wildfly_version>\server\startIndexAgent.cmd*.



Note: On Windows, if you start the index agent instance using the command, the status of the index agent service may not be accurately reflected in the Windows Management Console.

2. Start the index agent UI. In a browser, enter a URL: `http://<host:>/IndexAgent`
Where *<host>* is the DNS name of the machine on which you installed the index agent and *<port>* is the index agent port you specified during configuration (default: 9200).
3. 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.
4. 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. Only one index agent can actively reindex the repository.
 - **Continue: Reindex date time:** Continue indexing. *date* and *time* indicate when you stopped indexing.
5. If the index agent is running, choose one of these actions:
- **Detail Status:** Accumulated statistics since the last index agent restart and objects in the indexing queue.
 - **Refeed Tasks:** Displays all refeed tasks. You can pause, resume, and delete tasks, and export task details.
 - **Stop IA:** Halts the index agent. This does not stop the Windows service or the WildFly instance.
 - **IndexAgent:** 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. After submitting, refeed tasks are added to the Refeed Tasks page.
 - **Filters:** Displays all index agent filter information. You can add and delete filters.
 - **CS Status:** Displays Documentum Server information related to index agent.
 - **Logout:** Logs out of the index agent UI.

4.4 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 WildFly script or Windows service.

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

- *Windows:*

Stop the index agent service **Documentum <IndexAgent>** or run *xplore_home\<wildfly_version>\server\stopIndexAgent.cmd*.



Note: On Windows, if you stop the index agent instance using the command, the status of the index agent service may not be accurately reflected in the Windows Management Console.

- *Linux:*

Run *xplore_home/<wildfly_version>/server/stopIndexAgent.sh*.

4.5 Silent index agent startup

You can start or shut down the index agent through the index agent web application. You can also script the start in normal mode or shutdown using Documentum Server, iAPI or DFC. Starting in migration mode cannot be scripted.

Setting startup in Documentum Server

Set the `start_index_agents` parameter in `server.ini` to T.

At Documentum Server startup, the Server checks whether the index agent associated with the repository has started in normal mode. If not, and `start_index_agents` is T, the Server starts the index agent in normal mode using the `dm_FTIndexAgentBoot` job.



Note: The `dm_FTIndexAgentBoot` job uses the Index Agent Servlet and starts the index agent only if the WildFly instance has started.

When the index agent is configured in SSL mode and `start_index_agents = T` is set in `server.ini`, you must perform the following tasks to make sure that the index agent starts up along with the repository startup.

- Documentum Server must be run in SSL mode to make SSL handshake successful between Documentum Server and the WildFly instance of the index agent.
- When anonymous SSL (default one, without any certificates) is used, an anonymous cipher (such as, `TLS_DH_anon_WITH_AES_128_CBC_SHA` for AES-128) must be configured in `stand-alone.xml` of IndexAgent's WildFly instance.

Silent startup and shutdown with iAPI

Use the `retrieve` and `dump` commands to get the `index_name` attribute of the `dm_fulltext_index` object. You use this attribute value in the start or stop script. For example:

```
API> retrieve,c,dm_fulltext_index
...
3b0004d280000100
API> dump,c,1
...
USER ATTRIBUTES

index_name : Repo_ftindex_01
...
```

Now use the `retrieve` and `dump` commands to get the `object_name` attribute of the `dm_ftindex_agent_config` object. You use this attribute value in the start or stop script. For example:

```
retrieve,c,dm_ftindex_agent_config
...
0800277e80000e42
API> dump,c,1
```

```
...
USER ATTRIBUTES
object_name : Config13668VMO_9200_IndexAgent
```

Use the *apply* command to start or stop (shutdown) the index agent, and to view its current status. Syntax:

```
apply,c,,FTINDEX_AGENT_ADMIN,NAME,S,<index_name_of_dm_fulltext_index>,
AGENT_INSTANCE_NAME,S,<object_name_of_dm_ftindex_agent_config>,ACTION,
S,start|shutdown|status
```

The following example starts one index agent:

```
apply,c,NULL,FTINDEX_AGENT_ADMIN,NAME,S,LH1_ftindex_01,AGENT_INSTANCE_NAME,
S,Config13668VMO_9200_IndexAgent,ACTION,S,start
```

To start or stop all index agents, replace the index agent name with *all*. For example:

```
apply,c,NULL,FTINDEX_AGENT_ADMIN,NAME,S,LH1_ftindex_01,
AGENT_INSTANCE_NAME,S,all,ACTION,S,shutdown
```

Follow with these commands to get the results:

```
API> next,c,qNumber
...
OK
API> dump,c,qNumber
```

Where <Number> is the number of execution times that starts at 0 for the first command execution and increments by 1 with each execution.

Viewing the current index agent status returns one of the following:

- 0: The index agent is running.
- 100: The index agent has been shut down.
- 200: The index agent has a problem.

Setting startup with a list of file IDs

You can script startup to index a list of documents like the list generated by *ftintegrity*.

1. Start the index agent servlet in normal mode.
2. Create a text file with the object ID. Save as *ids.txt* in the WEB-INF/classes directory of *xplore_home/<wildfly_version>/server/DctmServer_IndexAgent/deployments/IndexAgent.war/*. (Specify the actual path to your index agent web application.)

The objects in *ids.txt* are automatically submitted for indexing. You can select the refeed task for these objects from the list of refeed tasks displayed from the index agent UI, and then export the task details to check whether the objects have been successfully indexed.



Note: After the refeed task has been created, the `ids.txt` will be removed from the WEB-INF/classes directory of `xplore_home/<wildfly_version>/server/DctmServer_IndexAgent/deployments/IndexAgent.war`.

Startup from the Java command line

Use the following command:

```
java com.documentum.server.impl.utils.IndexAgentCtrl -docbase_name
repositoryName -user_name userName -action actionPerformed
-index_agent instanceName
```

where `-action` argument value is one of the following: `start` | `shutdown` | `status` | `reset`.



Note: If you specify the instance name of IndexAgent, the specified action only takes effect on the specified IndexAgent; if no instance name of IndexAgent is specified, the specified action takes effect on all instances.

Silent startup and shutdown using DFC

The following method gets the `dm_fulltext_index` object, the `index_name` attribute, and sets the DQL query:

```
public void shutdownIA(IDfSession sess) throws DfException
{
    IDfPersistentObject FTIndexObj = (
        IDfPersistentObject) sess.getObjectByQualification(
            "dm_fulltext_index where is_standby = false");

    String indexName = FTIndexObj.getString("index_name");

    //Query definition
    String query = "NULL,FTINDEX_AGENT_ADMIN,NAME,S," +
        indexName + ",AGENT_INSTANCE_NAME,S,all,ACTION,S,shutdown";

    DfClientX clientX = new DfClientX();
    IDfQuery q = clientX.getQuery();

    q.setDQL(query);

    try
    {
        IDfCollection col = q.execute(sess, IDfQuery.DF_APPLY);
    }
    catch (DfException e)
    {
        e.printStackTrace();
    }
}
```

For startup, replace `shutdown` with `start` in the query definition.

4.6 Deleting an index agent

To delete an xPlore index agent:

1. Run `<xPlore_home>\setup\indexagent\configIndexagent.bat` (Windows) or `<xPlore_home>/setup/indexagent/configIndexagent.sh` (Linux).
2. Select **Delete Existing Indexagent** and choose the instance to delete.

If you are unable to connect to the repository, stop all WildFly processes in `xPlore_home`, and then manually delete the WildFly indexagent server instance.



Note: Deleting index agent does not automatically remove the corresponding xPlore domain. If you want to remove the xPlore domain, you must do it manually in xPlore Administrator.

Chapter 5

Deploying xPlore on Docker Compose Environment

5.1 Overview

Compose is a tool for defining and running multi-container Docker applications. With Compose, you can use a YAML file to configure the application's services. Then, with a single command, you can create and start all the services from your configuration.

5.2 Prerequisites

Ensure you complete the following activities before you deploy xPlore Docker images using Docker Compose:

- Download and configure the Docker from the Docker website.
- Download and configure the Docker Compose from the Docker website.
- Download the Documentum Server Images from OpenText My Support (<https://support.opentext.com>) and deploy them.
- Download the xPlore images (IndexServer, IndexAgent, CPS) from OpenText My Support.

5.3 Deploy xPlore Dockers using Docker Compose

5.3.1 Docker Compose Yaml Sample

Below is a sample of xPlore Docker compose yaml file, replace images and environment parameters with actual value and save it with name docker-compose.yaml before you use it.

This sample contains two instances of CPS, you can add or remove instances of CPS, but you must keep at least one CPS instance.

```
version: '3'
services:
  indexserver:
    image: indexserver #replace with your actual indexserver image tag here
    hostname: indexserver
    ports:
      - "9300:9300"
    volumes:
      - xplore:/root/xPlore/rtdata
  indexagent:
    image: indexagent #replace with your actual indexagent image tag here
    hostname: indexagent
    ports:
      - "9200:9200"
    environment:
      - ess_host=indexserver
```

```
- docbase_name=your_docbase_name #replace with actual value
- docbase_user=your_docbase_user_name #replace with actual value
- docbase_password=your_docbase_password #replace with actual value
- broker_host=your_doc_broker_host_ip_address #replace with actual value
- broker_port=your_doc_broker_host_port #replace with actual value
- registry_name=your_registry_name #replace with actual value
- registry_user=your_registry_user_name #replace with actual value
- registry_password=your_registry_password #replace with actual value
depends_on:
- indexserver
volumes:
- xplore:/root/xPlore/rtda
cps:
image: cps #replace with your actual cps image tag here
hostname: cps
environment:
- ess_host=indexserver
depends_on:
- indexserver
volumes:
- xplore:/root/xPlore/rtda
cps1:
image: cps #replace with your actual cps image tag here
hostname: cps1
environment:
- ess_host=indexserver
depends_on:
- indexserver
volumes:
- xplore:/root/xPlore/rtda
volumes:
xplore:
```

5.3.2 Start and Stop Docker Compose

5.3.2.1 Start Docker Compose

Change the directory location to the folder where you have saved the Docker compose file, and run the following command:

```
docker-compose up
```

You can visit the site, <http://<docker host machine IP>:9300/dsearchadmin> with the default credentials admin/password to check IndexServer and CPS status.

You can visit the site, <http://<docker host machine IP>:9200/IndexAgent> with the default credential docbase_user/docbase_password you set in IndexAgent environment section of the yaml file.

5.3.2.2 Stop Docker Compose

Run the following command to stop xPlore.

```
docker-compose down
```

Refer the Docker website for more details.

5.3.3 Configure SSL

To configure SSL of xPlore, you can simply do below items before you run the Docker compose.

- Change the port setting to 9302:9302 in `indexserver` section.
- Add environment parameter `config_ssl=true` in `indexserver` section.
- Change the port setting to 9202:9202 in `indexagent` section.
- Add 3 environment parameter `ess_port=9302, ess_protocol=https, config_ssl=true` in `indexagent` section.
- Add 3 envrionment parameter `ess_port=9302, ess_protocol=https, config_ssl=true` in `cps` section.

You can use the following URLs to check xPlore status:

- <https://<docker host machine IP>:9302/dsearchadmin>
- <https://<docker host machine IP>:9202/IndexAgent>

5.4 Environment Parameters

5.4.1 IndexServer

Parameter Name	Description
PASSWORD	<p>Default password of dtcm-users, dsearch, dsearch xdb, dsearch admin web console, and dsearch admin web console xdb.</p> <p>Default Value:password</p> <p>Required: No</p>
MEM-ARGS	<p>Part of <code>JAVA_OPTS</code>, bash script of starting indexserver.</p> <p><code>-XX:+UseCodeCacheFlushing -XX:ReservedCodeCacheSize=196m -Xms4096m -Xmx4096m -Xss1024k -XX:+UseCompressedOops -XX:+DisableExplicitGC -XX:-ReduceInitialCardMarks -Djava.awt.headless=true</code></p> <p>Default Value: copy default value to modify</p> <p>Required: No</p>

Parameter Name	Description
CONFIG_SSL	<p>To set the SSL of the IndexServer.</p> <p>Set to true to use SSL for the transport layer.</p> <p>Set to False to use the active normal mode, without SSL configuration.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
PERSIST_LOGS	<p>To persist the log files in <code>rtdata</code> volume or not.</p> <p>Log files including: <code>dsearch.log</code>, <code>dsearchadminweb.log</code>, <code>xdb.log</code>, <code>server.log</code>, <code>rest.log</code>, <code>cps.log</code> and <code>cps_daemon.log</code>.</p> <p>Set to true to keep log files in <code>/root/xPlore/rtdata/DctmServer_node/logs</code>.</p> <p>Set to false to keep log files in <code>/root/xPlore/logs</code>.</p> <p>Default Value: true</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
SCRIPT_DIR	<p>The directory which is used for custom script. User must mount custom scripts to this directory.</p> <p>Default Value: /root/xPlore/rtdata/script</p> <p>Required: No</p>
SCRIPT_INIT	<p>The bash script file which is under the <code>script_dir</code> directory is executed the first time the container is started, after the configuration.</p> <p>Default Value: init.sh</p> <p>Required: No</p>
SCRIPT_BEFORE_START	<p>The bash script file which is under the <code>script_dir</code> directory , is executed each time the container starts, before the JBoss start up.</p> <p>Default Value: before-start.sh</p> <p>Required: No</p>

Parameter Name	Description
EXT_CONF_PATH	<p>External configuration path which is used to load the external configuration files.</p> <p>Default Value: root/xPlore/external-configurations</p> <p>Possible or Recommended value: root/xPlore/external-configurations</p> <p>Required: No</p>
CONFIG_NEW_RELIC	<p>To check if configure new relic APM agent is present in the container. Set to True, the container automatically moves <code>newrelic.jar</code> (downloaded from the official website during the process of building image) to <code>\$NEW_RELIC_ROOT</code>. The container doesn't contain the <code>configure.yml</code>, file, you must load the <code>configure.yml</code> to <code>\$EXT_CONF_PATH</code> and when container is started, the <code>configuration.yml</code> file is automatically copied to <code>\$NEW_RELIC_ROOT</code>.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
NEW_RELIC_ROOT	<p>This is the path where the <code>newrelic.jar</code>, <code>newrelic.yml</code>, and the new relic log file are saved. It should not be set to <code>\$EXT_CONF_PATH</code>. <code>/root/xPlore/newrelic</code></p> <p>Required: No</p>
NEW_RELIC_APP_NAME_SUFFIX	<p>The suffix of new relic application name, for one xPlore instance (including IndexServer, IndexAgent, and cps). It is recommended to set the same suffix value, but for different xPlore instances, it must be set to a special and unique value to avoid conflict. The actual application name will add <code>\$NODE_NAME</code> (uppercase) as the prefix.</p> <p>Required: No</p>

5.4.2 IndexAgent

Parameter Name	Description
ESS_HOST	<p>Index Server hostname. It should be the Index Server service name in K8s.</p> <p>Default Value: indexserver</p> <p>Required: Yes</p>

Parameter Name	Description
ESS_PORT	<p>The port of the Index Server for communication. It can be set as 9300 or 9302. To use with SSL configuration, set this field to 9302.</p> <p>Default Value: 9300</p> <p>Possible or Recommended value: 9300 (normal), 9302 (ssl)</p> <p>Required: No</p>
ESS_PROTOCOL	<p>The communication protocol. It can be set as <code>http</code> or <code>https</code>. If using with SSL configuration, set this field to <code>https</code>.</p> <p>Default Value: http</p> <p>Possible or Recommended value: http (normal). https (ssl)</p> <p>Required: No</p>
MEM_ARGS	<p>Part of <code>JAVA_OPTS</code>, bash script that starts the indexagent.</p> <pre>-XX:+UseCodeCacheFlushing -XX:ReservedCodeCacheSize=196m -Xms256m -Xmx1024m -Xss256k -XX:+DisableExplicitGC</pre> <p>Possible or Recommended value: copy the default value to modify</p> <p>Required: No</p>
CONFIG_SSL	<p>To set the SSL of the IndexAgent. Set to <code>true</code> to use SSL for the transport layer. Set to <code>false</code> to use the normal mode, without SSL configuration.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
PERSIST_LOG	<p>To persist the log files in <code>rtdata</code> volume or not.</p> <p>Log files including: <code>dsearchclient.log</code>, <code>indexagent.log</code> and <code>server.log</code>.</p> <p>Set as <code>true</code> to keep log files in <code>/root/xPlore/rtdata/DctmServer_node/logs</code>; <code>false</code> to keep log files in <code>/root/xPlore/logs</code>.</p> <p>Default Value: true</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
DOCBASE_NAME	<p>Docbase name.</p> <p>Required: Yes</p>

Parameter Name	Description
DOCBASE_USE_R	Docbase user. Required: Yes
DOCBASE_PASWORD	Docbase password. Required: Yes
BROKER_HOST	Broker host.
BROKER_PORT	Broker port.
REGISTRY_NAME	Registry name.
REGISTRY_USE_R	Registry user.
REGISTRY_PASWORD	Registry password.
IS_STANDBY	Is a standby mode index agent. Default Value: false Possible or Recommended value: true/false Required: No
STANDBY_FULLTEXT_USER	Fulltext user in standby mode. Default Value: dm_fulltext_index_user_01 Required: No
STANDBY_FULLTEXT_INDEX_NAME	Fulltext index name in standby mode. Possible or Recommended value: \${docbase_name}_ftindex_00 Required: No
STANDBY_FULLTEXT_ENGINE_NAME	Fulltext engine name in standby mode. Possible or Recommended value: DSearch Fulltext Engine Configuration 00 Required: No
SCRIPT_DIR	The directory which is used for custom script. User should mount custom scripts to this directory. Default Value: /root/xPlore/rtdata/script Required: No

Parameter Name	Description
SCRIPT_INIT	<p>The bash script file that is under the directory <code>script_dir</code>. It is executed the first time the container starts after the configuration.</p> <p>Default Value: init.sh</p> <p>Required: No</p>
SCRIPT_BEFORE_START	<p>The bash script file that is under the directory <code>script_dir</code>. It is executed each time the container starts after the JBoss start up.</p> <p>Default Value: before-start.sh</p> <p>Required: No</p>
SCRIPT_AFTER_START	<p>The bash script file that is under the directory <code>script_dir</code>. It is executed each time the container starts before the JBoss start up.</p> <p>Default Value: after-start.sh</p> <p>Required: No</p>
FORCE_CREATE_OBJECT	<p>Force create the Documentum Server object and the Index Server domain. The default logic is, when set to <code>false</code>, it creates Documentum Server object and index server domain when no data is found under volume (<code>/root/xPlore/rtdata</code>). It will not create Documentum Server object and index server domain if the volume is not empty. Set it to <code>true</code> to override and force it to attempt to create objects.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
EXT_CONF_PATH	<p>External configuration path which is used to load the external configuration files. For external <code>dfc.properties</code>, only following properties are supported: <code>dfc.docbroker.host[N]</code> (<code>N=0, 1, 2....</code>) <code>dfc.docbroker.port[N]</code> <code>dfc.globalregistry.repository</code> <code>dfc.globalregistry.username</code>.</p> <p>Default Value: root/xPlore/external-configurations</p> <p>Possible or Recommended value: root/xPlore/external-configurations</p> <p>Required: No</p>

Parameter Name	Description
CONFIG_NEW_RELIC	<p>To check if the configure new relic APM agent is present in the container. If the value is set to true, the container will auto move <code>newrelic.jar</code> (which is downloaded from the official website during the process of building image) to <code>\$NEW_RELIC_ROOT</code>. The container doesn't supply the <code>configure.yml</code>, user should mount the <code>configure.yml</code> to <code>\$EXT_CONF_PATH</code> and when container is started, the <code>configuration.yml</code> is copied automatically to <code>\$NEW_RELIC_ROOT</code>.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
NEW_RELIC_ROOT	<p>The path in container to save <code>newrelic.jar</code> and <code>newrelic.yml</code>, and also the new relic log will be saved in this path. Ensure that it must not be <code>\$EXT_CONF_PATH</code>.</p> <p>Default Value: /root/xPlore/newrelic</p> <p>Required: No</p>
NEW_RELIC_APP_NAME_SUFFIX	<p>The suffix of new relic application name, for one xPlore instance (including IndexServer, IndexAgent, and cps). It is recommended to set the same suffix value. For different xPlore instances, it must be set to a special and unique value to avoid conflict. The actual application name will add <code>\$NODE_NAME</code> (uppercase) as its prefix.</p> <p>Required: No</p>

5.4.3 CPS

Parameter Name	Description
ESS_HOST	<p>Index Server hostname.</p> <p>Possible or Recommended value: indexserver</p> <p>Required: Yes</p>
ESS_PORT	<p>The port of the Index Server for communication.</p> <p>It can be set to 9300 or 9302.</p> <p>If SSL is configured, set this field to 9302.</p> <p>Default Value: 9300</p> <p>Possible or Recommended value: 9300 (normal) 9302 (ssl)</p> <p>Required: No</p>

Parameter Name	Description
ESS_PASSWORD	<p>The password of dsearch.</p> <p>Default Value: password</p> <p>Required: No</p>
ESS_PROTOCOL	<p>The communication protocol; it can be set to <code>http</code> or <code>https</code>.</p> <p>If SSL is configured, set this field to <code>https</code>.</p> <p>Default Value: http</p> <p>Possible or Recommended value: http (normal) https (ssl)</p> <p>Required: No</p>
MEM_ARGS	<p>Part of <code>JAVA_OPTS</code>, bash script for starting CPS.</p> <p>Default Value: -XX:+UseCodeCacheFlushing -XX:ReservedCodeCacheSize=196m -Xms4096m -Xmx4096m -Xss1024k -XX:+UseCompressedOops -XX: +DisableExplicitGC -XX:-ReduceInitialCardMarks -Djava.awt.headless=true</p> <p>Possible or Recommended value: copy default value to modify.</p> <p>Required: No</p>
CONFIG_SSL	<p>To set the SSL of the CPS.</p> <p>Set to <code>true</code> to use SSL for the transport layer.</p> <p>Set to <code>false</code> to the active normal mode, without SSL the configuration.</p> <p>It can be easily configured without setting any parameters with prefix as <code>ssl_</code>.</p> <p>All the parameters with the prefix <code>ssl_</code> use the default value.</p> <p>If you are not familiar with the configuration of SSL, you can skip defining the parameters.</p> <p>Default Value: false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>

Parameter Name	Description
PERSIST_LOGS	<p>To persist the log files in <code>rtdata</code> volume or not.</p> <p>Log files including: <code>cps_manager.log</code>, <code>cps_daemon.log</code>, and <code>server.log</code>.</p> <p>Set as <code>true</code> to keep the log files in <code>/root/xPlore/rtdata/DctmServer_node/logs</code>.</p> <p>Set as <code>false</code> to keep the log files in <code>/root/xPlore/logs</code>.</p> <p>Default Value: true</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
SCRIPT_DIR	<p>The directory that is used for custom script. User must mount custom scripts to this directory.</p> <p>Default Value: <code>/root/xPlore/rtdata/script</code></p> <p>Required: No</p>
SCRIPT_INIT	<p>The bash script file that is under the <code>script_dir</code>.</p> <p>It is executed the first time the container starts after the configuration.</p> <p>Default Value: <code>init.sh</code></p> <p>Required: No</p>
SCRIPT_BEFORE_START	<p>The bash script file that is under the <code>script_dir</code>.</p> <p>It is executed each time the container starts before the JBoss start up.</p> <p>Default Value: <code>before-start.sh</code></p> <p>Required: No</p>
SCRIPT_AFTER_START	<p>The bash script file that is under the directory <code>script_dir</code>.</p> <p>It is executed each time the container starts after the JBoss start up.</p> <p>Default Value: <code>after-start.sh</code></p> <p>Required: No</p>
EXT_CONF_PATH	<p>External configuration path which is used to mount external configuration files. <code>root/xPlore/external-configurations</code> <code>root/xPlore/external-configurations</code></p> <p>Required: No</p>

Parameter Name	Description
CONFIG_NEW_RELIC	<p>To check if configure new relic APM agent is present in the container. If the value is set to true, the container will automatically move the <code>newrelic.jar</code> (which is downloaded from the official website during the process of building image) to <code>\$NEW_RELIC_ROOT</code>. The container does not supply the <code>configure.yml</code>, the user must mount the <code>configure.yml</code> to <code>\$EXT_CONF_PATH</code>. When the container is started, the <code>configuration.yml</code> file is automatically copied to <code>\$NEW_RELIC_ROOT</code>. false</p> <p>Possible or Recommended value: true/false</p> <p>Required: No</p>
NEW_RELIC_ROOT	<p>The path in the container to save <code>newrelic.jar</code>, <code>newrelic.yml</code>, and the new relic log file. It must not be set to <code>\$EXT_CONF_PATH</code>. /root/xPlore/newrelic</p> <p>Required: No</p>
NEW_RELIC_APP_NAME_SUFFIX	<p>The suffix of new relic application name, for one xPlore instance (including IndexServer, IndexAgent, cps). It is recommended to set the same suffix value, but for different xPlore instances, it must be set to a special and unique value to avoid conflict. The actual application name will add <code>\$NODE_NAME</code> (uppercase) as prefix. No</p>

5.5 Limitations

- xPlore Docker images no longer supports multi-node deployment.
- xPlore Docker images no longer uses OIT and uses Tika instead as Text Extraction plugin.
- CPS component is mandatory. The local CPS in indexserver is only used for searching, while remote CPSs are used for indexing.
- Uploading a local file is not supported in Admin UI.

Chapter 6

Configuring WatchDog

6.1 Configuring DM_TICKET in WatchDog

To use DM_TICKET in WatchDog, the following configuration is required:

1. Open the \$xplore_home/watchdog/config/dfc.properties file. Create the file if it does not exist.
2. Copy the following entries from the dfc.properties file in the IndexAgent installation directory to the file from step 1, and then save the changes:
 - dfc.docbroker.host[0] = *host*
 - dfc.docbroker.port[0] = *port*
 - dfc.globalregistry.password = *password*
 - dfc.globalregistry.repository = *repository*
 - dfc.globalregistry.username = *username*

Chapter 7

Performing Silent Installation

7.1 Using response file to perform silent installation

xPlore supports silent installation as follows:

```
setup.exe -f "C:\software\winSilentInstall.properties"  
dsearchConfig.exe -f "C:\software\winSilentConfig.properties"
```

Use -r parameter to generate silent response file through GUI installation:

```
setup.exe -r winSilentInstall.properties  
dsearchConfig.exe LAX_VM "C:\xPlore\java64\JAVA_LINK\bin\java.exe"  
-r "C:\software\winSilentConfig.properties"
```

After the GUI installation is complete, change the value of INSTALLER_UI as silent and ensure PATH_TO_JAVA set to the installed JDK in the generated response file. Then use the -f parameter to perform silent installation.

```
setup.exe -f winSilentInstall.properties  
dsearchConfig.exe LAX_VM "C:\xPlore\java64\JAVA_LINK\bin\java.exe"  
- f "C:\software\winSilentConfig.properties"
```

7.2 Setting up silent installation

The silent installation scripts install xPlore primary, secondary, and spare instances on local or remote hosts. You can also create a new index agent for each repository on local or remote hosts.



Note: Silent installation does not support the following index agent configurations:

- Configuring an index agent to communicate with Documentum Server using non-anonymous or certificate-based SSL secure connections.
- Creating an index agent for a standby xPlore deployment in an HA configuration.

For these configurations, you must manually configure the index agent. See “[Configuring the index agent](#)” on page 45.

For remote installation, a third-party tool is required:

- Windows: Pstools, downloaded from Microsoft.
- Linux: Expect, included in the silent installation archive.
- Documentum Server: Configure at least one Documentum repository and one connection broker.

- Linux: Installation of xPlore in console mode is not supported. Set up your environment to support GUI installation. Installation on 64-bit Linux requires the following RPM packages before xPlore installation:
 - libXp-<version>-<release>. <architecture> (e.g. libXp-1.0.0-15.1.el6.i686)
 - libXi-<version>-<release>. <architecture> (e.g. libXi-1.3-3.el6.i686)
 - libXtst-<version>-<release>. <architecture> (e.g. libXtst-1.0.99.2-3.el6.i686)
 - libXt-<version>-<release>. <architecture> (e.g. libXext-1.1-3.el6.i686)

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

All instances must be installed on the same operating system (Windows or Linux).

- To avoid permission-related errors, the same installation owner (same username, same password, and same Windows domain) must install all instances.
1. Extract the silent installation archive SilentInstaller.zip to a directory. SilentInstaller.zip is a separate download. The base directory is now SilentInstaller.
 2. Download and install the supported Oracle JDK 11 or OpenJDK 11 version from Oracle website. OpenJDK documentation contains detailed information.
 3. Create a *jdk* subdirectory of SilentInstaller and extract the JDK to this directory.
 4. Download ant version 1.7 or higher and place it in a subdirectory of SilentInstaller named *ant*. Make sure your ant library contains ant-contrib.jar. Download it if it is missing, and place it in the *ant/lib* directory.
 - If you are installing multiple hosts on Linux, then before you start silent installation ensure that you install the required packages, *tcl* and *expect*.



- Note:** On an older version of Linux platform, if you encounter installation issues, follow the instructions provided in the *tcl-expect-install.txt* file located at *SilentInstaller/3rdParty/Expect*.
- If you are installing to multiple hosts on Windows, add the *pstools* directory to *SilentInstaller/3rdParty*. You can download *pstools* from <http://technet.microsoft.com>.

Your silent install directory must have the following content:

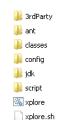


Figure 7-1: Silent install directory

5. Download the xPlore installer to a temporary directory and note the location. Supply this path as the value of *installers.location* when you configure silent installation.

7.3 Configuring silent installation

You can configure one or more xPlore instances and zero or more index agents. Each Documentum repository can have only one index agent. You can configure an additional index agent to process ACLs and groups. For instructions, see *OpenText Documentum xPlore Administration and Development Guide*.

1. Edit the file `silent.properties` in *SilentInstaller/config*. You do not need to edit any other properties files in this directory.
2. Host details:
 - *SMTP_HOST*: DNS name of SMTP host (not Microsoft Outlook server). If blank, no watchdog notifications are sent to the administrator.
 - *ADMINISTRATOR_EMAIL_ADDRESS*: Email address of xPlore administrator, to receive instance watchdog notifications. If blank, no watchdog notifications are sent to the administrator.
 - *installers.location*: Location of the xPlore expanded installer archive containing `setup.exe` or `setup.sh`.
 - *common.installLocation*: The path on each host machine for the xPlore home directory (*xplore_home* in xPlore documentation). A subdirectory is created in this path to serve as the temporary staging location for content to be indexed.
3. Storage location: Installs a data and config directory on the specified host. The index and configuration files do not have to be stored on the same host as the primary instance. The storage host and parent folder must be accessible by all xPlore instances.
 - *storage.fqdn*: Fully qualified domain name for storage host. If specified, *storage.ip* is not used. Must contain only alphanumeric characters.
 - *storage.ip*: IP address of the host where the index data will be stored.
 - *storage.parent.folder*: Path on the storage host that will contain the data directory.
4. Edit the primary instance settings.
 - *primary.instance.fqdn*: Fully qualified domain name of xPlore primary instance. If specified, *primary.instance.ip* is not used. Must contain only alphanumeric characters.
 - *primary.instance.ip*: IP address of the xPlore primary instance.
 - *primary.instance.port*: Base port for the primary instance. Use this port to open xPlore administrator, for example, `http://server.opentext.org:9300/dsearchadmin`. An additional 99 ports are used by xPlore.

- *primary.instance.name*: Name that identifies the instance. Use a name that tells you this is a primary instance. The instance name must contain alphanumeric characters.
 - *primary.instance.password*: Specify the password for the xDB administrator. Use the same password for all xPlore instances. The user must have full permissions on the directory specified in installers.location . The password is not encrypted. (It is encrypted in the GUI installer.)
5. Edit admin username and password. They are used to log in to the xPlore Administrator console as superuser.
 - *admin.username*: Name of default admin user.
 - *admin.password*: Password for the default admin user. The password is not encrypted (it is encrypted in the GUI installer).
 6. Edit the secondary instance settings. Each instance has the same keys as the primary instance, in the form nonPrimary.instance.key_name.N where N is a unique integer. If you are installing only one instance, you will supply this number N as a command line argument.
 - *nonPrimary.count*: Number of secondary instances already installed in the xPlore federation. If you are not installing secondary instances, set to 0.
 - If the secondary instance is on the same host as the primary, choose a port at least 100 units different from the primary port and any other secondary ports.
 - If the secondary instance is to be used as a spare for failover, set the value of nonPrimaryInstance.instance.isSpareNode.N to 1.
 7. Edit the index agent instance settings. Do not modify configIA.properties: it will be updated by the values that you enter in silent.properties. Each instance has keys in the form indexagent.instance.key_name.N where N is a unique integer. If you are installing only one index agent instance, you will supply this number N as a command line argument. Each *indexagent.docbase.name* must be unique.
 - *indexagent.count*: Number of existing index agents already installed.
 - *indexagent.instance.name*: Unique name for the index agent. Configure only one index agent per repository.
 - *indexagent.instance.fqdn*: Fully qualified domain name of index agent host. If specified, indexagent.instance.ip is not used. Must contain only alphanumeric characters.
 - *indexagent.instance.ip*: IP address of the index agent host.
 - *indexagent.instance.port*: Unique port on the index agent host. The index agent uses 20 additional consecutive ports.
 - *indexagent.instance.password*: Password for the index agent administrator. Can be the same as the xPlore administrator password.

- *indexagent.docbase.name*: Repository served by this index agent. Each repository must have its own index agent. The name must contain alphanumeric characters.
- *indexagent.docbase.user*: Specify the Documentum Server instance owner.
- *indexagent.docbase.password*: Specify the password for the Documentum Server instance owner.
- *indexagent.connectionBroker.host*: IP address of the Documentum Server connection broker host.
- *indexagent.connectionBroker.port*: Port number for the Documentum Server connection broker. Default: 1489.
- *indexagent.globalRegistryRepository.name*: Name of a global registry repository that projects to the connection broker specified above.
- *indexagent.globalRegistryRepository.user*: Name of the global registry user. Default: dm_bof_registry
- *indexagent.globalRegistryRepository.password*: Password for the global registry user.
- *indexagent.localcontent.area*: Temporary location for index agent staging of content. This location is not the same as *indexagent.instance.storage.name*.

7.4 Running the silent install script

- Run xplore.bat or xplore.sh in the SilentInstaller directory. For local xPlore installations, OpenText highly recommends that you use the target silent.install when running the script:

```
xplore.bat silent.install
```

or

```
xplore.sh silent.install
```

- Add an xPlore secondary instance. Configure all existing instances in silent.properties. Add the nonPrimary.instance settings and set nonPrimary.count to the number of existing secondary instances. The following example runs the script to add a third secondary instance. That instance is configured in silent.properties as nonPrimary.instance.key_name.3:

```
xplore.bat|sh config.one.instance -Dname=dsearch -Dnumber=3
```

- Add an index agent instance. Configure all existing instances in silent.properties. Add the indexagent.instance settings and set indexagent.count to the number of existing index agent instances. The following example runs the script to add a third index agent. This index agent serves a different repository than the other two index agents. The new indexagent instance is configured in silent.properties as indexagent.key_name.3:

```
xplore.bat|sh config.one.instance -Dname=indexagent -Dnumber=3
```

The installation log `install.log` is written to the directory specified in `common.installLocation`. If you install to multiple instances, an install log is written

to the install location on each instance. A successful script output is like the following:

```
C:\Users\Administrator\Downloads\SilentInstaller>xplore.bat silent.install
Buildfile: C:\Users\Administrator\Downloads\SilentInstaller\script\build.xml

validate:
[echo] validation begins
[echo] output: false
[echo] validation completes
[delete] Deleting: C:\Users\Administrator\Downloads\SilentInstaller\recorder.txt

silent.install:

setup:
[copy] Copying 1 file to C:\Users\Administrator\Downloads\win64
[echo] begin == setup
[echo] windows install
[echo] end == setup
[echo] waiting for xPlore setup ...
[echo] end == waiting
[delete] Deleting: C:\Users\Administrator\Downloads\win64\silent.properties

primary.config.update:
[echo] begin to update primary configuration

set.data.folder:
[echo] xplore data folder is: C:\xPlore\storage\data

set.config.folder:
[echo] xplore config folder is: C:\xPlore\storage\config

set.whichone:
[echo] primary.fqdn.or.ip is 192.168.2.104
[echo] primary.fqdn.or.ip use 192.168.2.104
[propertyfile] Updating property file: C:\Users\Administrator\Downloads\SilentInstaller\config\config.properties
[echo] C:\xPlore\storage\data
[echo] end to update primary configuration

config.dsearch:
[move] Moving 1 file to C:\xPlore\setup\dsearch
[copy] Copying 1 file to C:\xPlore\setup\dsearch
[echo] begin == configure dsearch
[exec]
[exec] C:\xPlore\setup\dsearch>dsearchConfig.exe LAX_VM "C:\xPlore\java64\JAVA_LINK\bini\java.exe" -f config.properties
[echo] end == configure dsearch
[delete] Deleting: C:\xPlore\setup\dsearch\config.properties
[move] Moving 1 file to C:\xPlore\setup\dsearch

start.primaryInstance:
[echo] begin == start primary instance, please wait ...

waitfor.primary.started:
[echo] begin == wait for dsearch (instances) service started
[echo] end waitfor
[echo] end == start primary instance
[echo] user wants to install 1 non-primary instances

local.config.nonPrimary:
[echo] there are 1 nonPrimary instances needed to be configured
[echo] begin to configure nonPrimary instances
[echo] start to configure the 1th non-primary instance

set.data.folder:
[echo] xplore data folder is: C:\xPlore\storage\data

set.config.folder:
[echo] xplore config folder is: C:\xPlore\storage\config
```

```

set.whichone:
[echo] primary.fqdn.or.ip is 192.168.2.104
[echo] primary.fqdn.or.ip use 192.168.2.104
[propertyfile] Updating property file: C:\Users\Administrator\Downloads\SilentInstaller\config\config.properties

config.dsearch:
[move] Moving 1 file to C:\xPlore\setup\dsearch
[copy] Copying 1 file to C:\xPlore\setup\dsearch
[echo] begin == configure dsearch
[exec]
[exec] C:\xPlore\setup\dsearch>dsearchConfig.exe LAX_VM "C:\xPlore\java64\JAVA_LINK
\bin\java.exe" -f config.properties
[echo] end == configure dsearch
[delete] Deleting: C:\xPlore\setup\dsearch\config.properties
[move] Moving 1 file to C:\xPlore\setup\dsearch
[echo] finish to configure non-primary instance: DsearchNode2

start.nonPrimaryInstance:
[echo] begin == start nonPrimary instance
[echo] begin == wait for nonPrimary instance DsearchNode2 started
[echo] end waitfor
[echo] end == start nonPrimary instance DsearchNode2
[echo] end to configure nonPrimary instance

local.config.indexagent:

set.whichone:
[echo] primary.fqdn.or.ip is 192.168.2.104
[echo] primary.fqdn.or.ip use 192.168.2.104
[echo] xplore index agent content area is: C:\xPlore\IndexAgent
[propertyfile] Updating property file: C:\Users\Administrator\Downloads\SilentInstaller\config\configIA.properties

config.one.indexagent:

waitfor.primary.started:
[echo] begin == wait for dsearch (instances) service started
[echo] end waitfor
[copy] Copying 1 file to C:\xPlore\setup\indexagent
[copy] Copying 1 file to C:\xPlore\setup\indexagent
[echo] begin == config index agent
[exec]
[exec] C:\xPlore\setup\indexagent>iaConfig.exe LAX_VM "C:\xPlore\java64\JAVA_LINK
\bin\java.exe" -f config.properties
[echo] end == config index agent
[delete] Deleting: C:\xPlore\setup\indexagent\config.properties
[copy] Copying 1 file to C:\xPlore\setup\indexagent
[echo] finish to configure the index agent instance: IndexAgent

start.indexagent:
[echo] begin == start index agent, please wait ...
[echo] end == start index agent

BUILD SUCCESSFUL
Total time: 4 minutes 35 seconds

```

7.5 Verifying silent installation

After successful installation, the silent installation script starts the instances.

1. To test whether the primary instance is running, open your web browser and enter the following URL: `http://host:port/dsearch`. *host* is the DNS name of the primary instance host and *port* is the xPlore port (default 9300). These two settings are configured in your silent.properties file. For example:

```
http://server.opentext.org:9300/dsearch
```

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

```
The xPlore instance PrimaryDsearch [version=current_version] normal
```

2. Start xPlore administrator with a URL to the primary instance. Substitute the primary instance name and port: `http://host:port/dsearchadmin`. For example:

```
http://server.opentext.org:9300/dsearchadmin
```

3. To start indexing or migration, open the index agent UI with the following URL: `http://host:IA_port/IndexAgent/started_dss.jsp`. For example:

```
http://server.opentext.org:9200/IndexAgent/started_dss.jsp
```

7.6 Troubleshooting silent installation

Secondary instance failure

If you use different passwords for secondary instances in your silent installation properties, you see errors such as com.xhive.error.XhiveException: FEDERATION_NOT_FOUND in xdb.log or Logon failure: unknown user name or bad password in cps.log. The log files are located in the logs subdirectory of the WildFly deployment directory. Perform the following steps to rectify:

1. Stop the primary instance.
2. Edit `indexserver-bootstrap.properties` of the secondary instance `xplore_home/<wildfly_version>/server/DctmServer_PrimaryDsearch/deployments/dsearch.war/WEB-INF/classes`. Change the `adminuser-password` to that of the primary instance, in clear text. When you restart, that password is encrypted using a FIPS 140-2 validated encryption module..
3. Restart the primary instance and then the secondary instance.

7.7 Sample single-instance installation

The following silent.properties file configures a single secondary instance. The IP address of the primary and new instance hosts are the same.

Run a single-instance installation on the command line with the config.one.instance command. For the -Dnumber argument, specify the count index. For example, the following command installs the instance configured as *nonPrimary.instance.ip.1=10.8.46.133*:

```
xplore.bat config.one.instance -Dname=dsearch -Dnumber=1
```

silent.properties:

```
SMTP_HOST=mailhub.opentext.com
ADMINISTRATOR_EMAIL_ADDRESS=aa@opentext.com
common.installLocation=C:\\xPlore

storage.ip=10.8.46.183
storage.parent.folder=C:\\xPlore

xPlore.installation.owner.domain=
xPlore.installation.owner.username=Administrator
xPlore.installation.owner.password=password

primary.instance.ip=10.8.46.133
primary.instance.hostname=Config8518VM0
primary.instance.port=9300
primary.instance.name=PrimaryDsearch
primary.instance.password=password

admin.username=admin
admin.password=password

#=====configure secondary instance=====
nonPrimary.count=0
# info for the first secondary instance
nonPrimary.instance.ip.1=10.8.46.133
nonPrimary.instance.hostname.1=Config8518VM0
nonPrimary.instance.port.1=9400
nonPrimary.instance.name.1=DsearchInstance1
# If this is a spare instance, set following value to 1 instead of 0 (not spare)
nonPrimary.instance.isSpareNode.1=0

nonPrimary.instance.ip.2=10.37.10.133
nonPrimary.instance.hostname.2=HostB
nonPrimary.instance.port.2=9500
nonPrimary.instance.name.2=DsearchInstance2
# If this is a spare instance, set following value to 1 instead of 0 (not spare)
nonPrimary.instance.isSpareNode.2=0

nonPrimary.instance.ip.3=10.37.10.103
nonPrimary.instance.hostname.3=HostB
nonPrimary.instance.port.3=9600
nonPrimary.instance.name.3=DsearchInstance3
# If this is a spare instance, set following value to 1 instead of 0 (not spare)
nonPrimary.instance.isSpareNode.3=0

nonPrimary.instance.ip.4=10.37.10.103
nonPrimary.instance.hostname.4=HostB
nonPrimary.instance.port.4=9700
nonPrimary.instance.name.4=DsearchInstance4
# If this is a spare instance, set following value to 1 instead of 0 (not spare)
nonPrimary.instance.isSpareNode.4=0

#=====configure the index agent=====
```

```
# Specify the number of index agents to be configured.  
# If you don't need to install an index agent, set to 0  
indexagent.count=1  
  
# Info for the first index agent, index begins with 1  
# To configure a second index agent for a different repository,  
# copy these keys and change index 1 to 2.  
indexagent.instance.name.1=IndexAgent  
indexagent.instance.ip.1=10.37.10.102  
indexagent.instance.hostname.1=HostA  
indexagent.instance.port.1=9200  
indexagent.instance.password.1=password  
indexagent.docbase.name.1=repository  
indexagent.docbase.user.1=root  
indexagent.docbase.password.1=password  
indexagent.connectionBroker.host.1=10.37.10.83  
indexagent.connectionBroker.port.1=1489  
indexagent.globalRegistryRepository.name.1=repository  
indexagent.globalRegistryRepository.user.1=dm_bof_registry  
indexagent.globalRegistryRepository.password.1=password  
  
indexagent.instance.name.2=IndexAgent2  
indexagent.instance.ip.2=10.37.10.102  
indexagent.instance.hostname.2=HostA  
indexagent.instance.port.2=9100  
indexagent.instance.password.2=password  
indexagent.docbase.name.2=repository  
indexagent.docbase.user.2=root  
indexagent.docbase.password.2=password  
indexagent.connectionBroker.host.2=10.37.10.7  
indexagent.connectionBroker.port.2=1489  
indexagent.globalRegistryRepository.name.2=repository  
indexagent.globalRegistryRepository.user.2=dm_bof_registry  
indexagent.globalRegistryRepository.password.2=password
```

Chapter 8

Setting up High Availability

8.1 Spare instance HA

The following diagram illustrates a Documentum Server HA deployment coupled with an xPlore spare instance HA. A single spare instance can support multiple xPlore instances. Your risk increases in proportion to the number of running xPlore instances that a single spare instance supports. You must perform more manual tasks when replacing a primary instance than when replacing a secondary instance. For instructions on configuring and activating an xPlore spare instance, see the *OpenText Documentum xPlore Administration and Development Guide*.

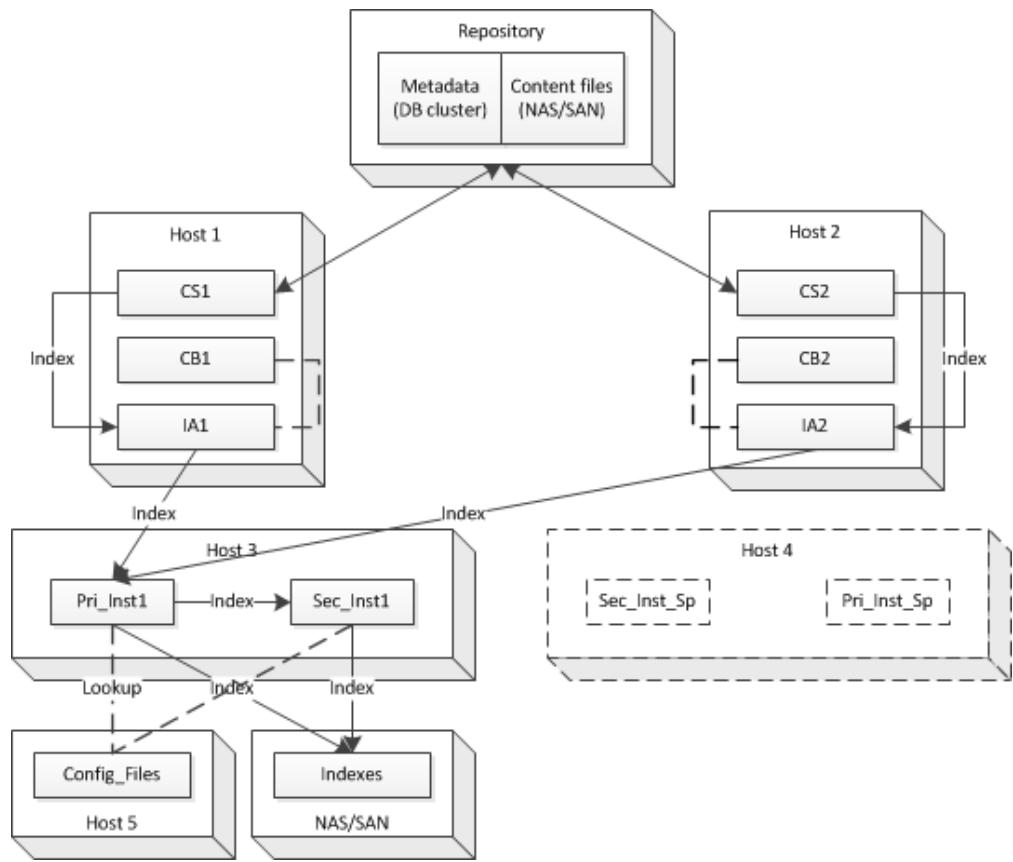


Figure 8-1: xPlore spare instance HA before failover

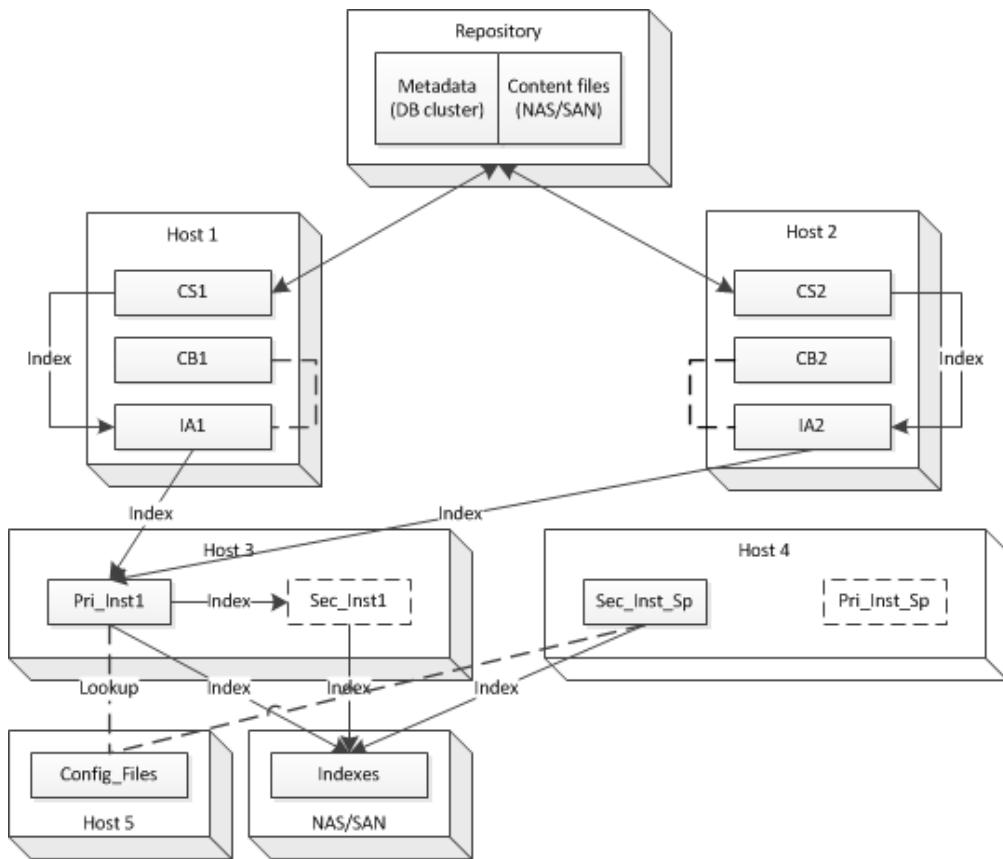


Figure 8-2: xPlore spare instance HA after failover

Key:

- **Host1:** Primary Documentum Server host
 - Documentum Server (CS1)
 - Connection broker (CB1)
 - Index agent1 (IA1)
- **Host2:** Secondary Documentum Server host
 - Documentum Server (CS2)
 - Connection broker (CB2)
 - Index agent2 (IA2)
- **Host3:** Primary xPlore host:
 - Primary instance (Pri_Inst)
 - Secondary instance (Sec_Inst)

- Host4: Spare xPlore host
 - Spare primary instance (Pri_Inst_Sp)
 - Spare secondary instance (Sec_Inst_Sp)
- Host5: xPlore primary instance data, log, and configuration files, which must be writable from both the running instances as well as the spare instances. When you start the spare instances on Host4 to replace Host3, they must load the same configuration parameters as the instances on Host3.
- NAS/SAN: Full-text indexes

For indexing to be routed to the correct Documentum Server, IA1 must connect to CB1, and IA2 must connect to CB2. You specify the connection broker when you configure the index agent.

After failover from Host3, you manually start the Sec_Inst_Sp on Host4, which reads the configuration data from Host5. All indexing now goes through Sec_Inst_Sp on Host4.

If the entire Host3 machine fails, switch to both Pri_Inst_Sp and Sec_Inst_Sp on Host4. For instructions on how to switch, see *OpenText Documentum xPlore Administration and Development Guide*.

8.2 Configuring active-active load balancer support

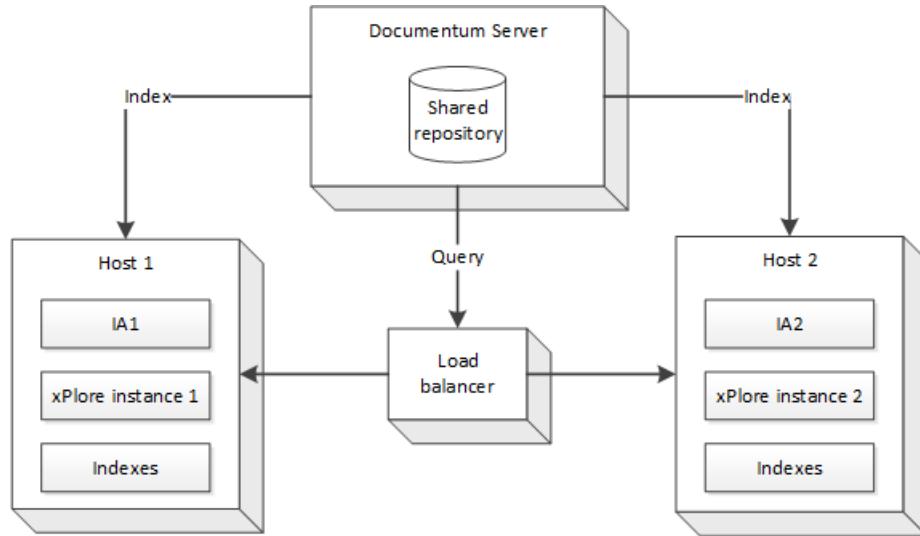
An active-active load balancer configuration has the shortest down time of all high availability strategies. The load balancer automatically handles failover. This configuration places a greater load on the Documentum Server because duplicate full-text queue items are created and handled.

Two separate xPlore instances and two separate index agents are deployed against the same Documentum Server and repository. For indexing, the separate index agents keep indexes separately up to date. The load balancer distributes queries.

This deployment gives both indexing and searching high availability. If indexing stops on one xPlore instance, the queue items for that instance accumulate. The corresponding queue items for the other instance continue to be processed. When the down instance resumes, it processes the accumulated queue items and catches up with the other instance.



Note: The two indexes may not be identical if one xPlore instance is catching up due to a disruption in indexing. In this case, the same query can get different results when it is sent to one or the other xPlore installation.

**Figure 8-3: Load balancer active-active strategy**

In this diagram, separate index agents each handles the same indexing queue items. Separate xPlore primary instances handle the indexing, and the resulting indexes are approximately identical. The load balancer handles queries from Documentum Server and distributes the queries to an available xPlore instance.

The generic procedure is as follows:

1. Install and configure:
 - a. The Documentum Server with a repository
 - b. The load balancer
 - c. The first xPlore primary instance and index agent
2. Stop the index agent for the first xPlore primary instance.
3. Install and configure the second xPlore primary instance and the index agent, select **Create a new Indexagent to standby xPlore Server** when configuring the index agent.
4. Restart the Documentum Server and verify that two dm_fulltext_index, two dm_ftengine_config, and two dm_ftindex_agent_config objects have been created:

```

API> ?,c,select r_object_id,index_name,ft_engine_id,is_standby from
dm_fulltext_index

API> ?,c,select r_object_id,object_name from dm_ftengine_config

API> ?,c,select r_object_id,object_name,index_name,queue_user from
dm_ftindex_agent_config

```

The output of each command should return two objects.

5. Start xPlore instances and index agents on both hosts.

6. Verify the heartbeat (keepalive) URL for the load balancer. This URL detects failure of xPlore instances and returns true with a response code of 257 when the instance is searchable:

```
http://<xPlore_host>:<xPlore_port>/dsearch/?action=CheckSearchable
```

7. Update dsearch_qrserver_host and dsearch_qrserver_port values of both dm_ftengine_config objects in the repository with the VIP (Virtual IP) and the port used by the load balancer. Both query plugins point to the same load balancer IP address and port. For example:

```
API> ?,c,select r_object_id,object_name from dm_ftengine_config
r_object_id          object_name
-----
0801b6698001aacb    DSearch Fulltext Engine Configuration 00
0801b6698001b110    DSearch Fulltext Engine Configuration
(2 rows affected)

API>fetch,c, 0801b6698001aacb
API>get,c,1, param_name[3]
dsearch_qrserver_host
API>set,c,1, param_value[3]
<vip used by load balancer for xPlore HA deployment>
API>get,c,1, param_name[5]
dsearch_qrserver_port
API>set,c,1,param_value[5]
<port used by load balancer for xPlore HA deployment>
API>save,c,1
API>fetch,c, 0801b6698001b110
API>get,c,1, param_name[3]
dsearch_qrserver_host
API>set,c,1, param_value[3]
<vip used by load balancer for xPlore HA deployment>
API>get,c,1, param_name[5]
dsearch_qrserver_port
API>set,c,1,param_value[5]
<port used by load balancer for xPlore HA deployment>
API>save,c,1
```

8. Enable load balancing by setting the load_balancer_enabled parameter to true for both the dm_ftengine_config objects.

```
API> ?,c,select r_object_id,object_name from dm_ftengine_config
r_object_id          object_name
-----
0801b6698001aacb    DSearch Fulltext Engine Configuration 00
0801b6698001b110    DSearch Fulltext Engine Configuration
(2 rows affected)

API>fetch,c,0801b6698001aacb
API>append,c,1,param_name
SET>load_balancer_enabled
API>append,c,1,param_value
SET>true
API>save,c,1

API>fetch,c, 0801b6698001b110
API>append,c,1,param_name
SET>load_balancer_enabled
API>append,c,1,param_value
SET>true
API>save,c,1
```

9. Restart the index agents and the Documentum Server.

The following URL displays True when the host is searchable:

```
http://<LB vip address>:<LB port>/dsearch/?action=CheckSearchable
```

8.3 Setting up primary instance HA (active-active)

Ensure that no users are connected to the repository. Upgrade Documentum Servers to a supported version for xPlore.

For all scripts in the following procedure:

- Scripts are located in:
 - Windows: %DM_HOME%\install\admin
 - Linux: \$DM_HOME/install/admin
- Enter a space before and after the double hyphen.
- No environment variables in paths.

This procedure sets up a single repository with HA (two or more Documentum Server hosts). If you are setting up a single repository with a single Documentum Server host, follow these steps but specify the same connection broker for both index agents.

1. If you are upgrading an existing Documentum Server 5.3 SP2 or 5.3 SP3 high-availability deployment, run the following scripts to create HA objects in the repository before you install xPlore and the xPlore index agent:
 - a. Delete existing index agents and index servers.
 - b. Execute the `create_fulltext_objects_ha.ebs` script with the `HACleanupBeforeUpgradeStep`. This script deletes all `dm_ftengine_config` and `dm_fulltext_index` objects.

Syntax:

```
dmbasic -f create_fulltext_objects_ha.ebs -e HACleanupBeforeUpgradeStep --<repository><superuser><password>
```

2. Install and configure the primary xPlore deployment (see “[Installing the xPlore primary instance with Documentum Server 7.3 and later](#)” on page 26) and primary xPlore index agent (see “[Configuring the index agent](#)” on page 45). Specify the primary Documentum Server’s connection broker as the connection broker for this index agent. The index agent connects to the repository and creates the full-text indexing objects that are required to run the scripts in the subsequent steps.
3. Install and configure a second primary xPlore deployment on another host (see “[Installing the xPlore primary instance with Documentum Server 7.3 and later](#)” on page 26.) Also install and configure a standby Index Agent on this host (see “[Configuring the index agent](#)” on page 45.) Specify the secondary Documentum Server’s connection broker as the connection broker for this Index Agent.

4. Use iAPI to confirm that the scripts created two of the following objects, one for each Documentum Server:

```
dm_fulltext_index
dm_ftengine_config
dm_ftindex_agent_config
```

- a. Each `dm_fulltext_index` object is associated with one xPlore index. Each has an associated `dm_ftengine_config` object. To verify `dm_fulltext_index` for the default Documentum Server:

```
API> ?,c,select e.r_object_id,e.object_name from dm_ftengine_config e,
dm_fulltext_index fi where e.r_object_id=fi.ft_engine_id and fi.is_standby=0
```

To verify on the standby Documentum Server:

```
API> ?,c,select e.r_object_id,e.object_name from dm_ftengine_config e,
dm_fulltext_index fi where e.r_object_id=fi.ft_engine_id and fi.is_standby=1
```

- b. To verify that there are `dm_ftindex_agent_config` objects for each index and that the queue user is correctly set for the second index, execute the following Documentum API queries. For the default index agent:

```
API> ?,c,select ia.r_object_id,ia.object_name,ia.queue_user from
dm_ftindex_agent_config ia,dm_fulltext_index fi where
ia.index_name=fi.index_name and fi.is_standby=0
```

For the standby index agent:

```
API> ?,c,select ia.r_object_id,ia.object_name,ia.queue_user from
dm_ftindex_agent_config ia,dm_fulltext_index fi where
ia.index_name=fi.index_name and fi.is_standby=1
```



Note: The `object_name` is the name of the index agent and `index_name` is the name of the index.

5. Start both index agents. The indexes are created.

To verify that the indexes have been created correctly, run `ftintegrity` or the `State of the Index` job on each Documentum Server. (The `State of the Index` job is available on Documentum Server 6.7 or higher.) See the *OpenText Documentum xPlore Administration and Development Guide*.

8.4 Enabling queries after active-active failover

When the primary xPlore deployment fails, indexing continues on the secondary xPlore deployment. You must manually enable queries to be serviced by the secondary xPlore deployment.

1. Retrieve the secondary `dm_fulltext_index` object ID with this DQL statement:

```
select ft_engine_id from dm_fulltext_index where is_standby=1
```

2. On both the primary and secondary Documentum Server, edit the `server.ini` file located in `documentum_home/dba/config/<servername>`. Under `[SERVER_STARTUP]`, set the `ftengine_to_use` value to the secondary `dm_fulltext_index` object ID.

3. Restart both Documentum Servers.

8.5 Restoring data after a deployment fails

In HA setup, you can restore data from one xPlore deployment to another when one of the deployments fails.

The following steps use two example deployments: DeploymentA and DeploymentB. xPlore is set up on these two Deployments with mirrored information: same folder structures, same username/password, same instance name, same ports, etc. The example assumes that data corruption occurs on DeploymentB.

1. Stop xPlore and IndexAgent on DeploymentA. Record the time the index agent is shut down.
2. Stop xPlore and IndexAgent on DeploymentB.
3. Backup `xplore_home/config/indexserverconfig.xml` and `xplore_home/config/XhiveDatabase.bootstrap` on DeploymentB. You will use this backup to recover configuration information after you restore data.
4. Copy the directories `xplore_home/data` and `xplore_home/config` from DeploymentA to DeploymentB. Do not change file permission.
5. Optionally, start xPlore and the index agent in normal mode (not migration mode) on DeploymentA to put DeploymentA in service.
6. Update `xplore_home/config/indexserverconfig.xml` on DeploymentB with settings in the original backup copy. Change the attributes “url” and “hostname” of the XML element `index-server-configuration.node` to match those values in the backup file.

For Windows systems that use UNC paths, update the following XML elements or attributes to their original values in the backup copy:

- `/storage-locations/storage-location[@path]`
 - `/admin-config/backup-location[@path]`
7. Update `xplore_home/config/XhiveDatabase.bootstrap` on DeploymentB to match the original file. Change attribute “host” of the element `/server/node` to match the value in the backup file.

For Windows systems that use UNC paths, update the following XML elements or attributes to their original values in the backup copy. Update the path only and do not change the file name.

- `/server/node/log/[@path]`
 - `/server/database/segment/file/[@path]`
8. If DeploymentB failed some time ago, you must clear many duplicate indexing tasks that are queued as `dmi_queue_items`. Run DQL to truncate the queued items for IndexAgentB: In the following example, `IA_user` is the user for

IndexAgentB, *date_value* is a time just before the index agent shutdown time recorded in step1.

```
? ,c,delete dmi_queue_item objects where name='IA_user' and  
date_sent<=date('date_value')
```

For example:

```
? ,c,delete dmi_queue_item objects where name='dm_fulltext_index_user'  
and date_sent<=date('5/22/2012 21:24:45')
```

9. Start xPlore and the index agent in normal mode on DeploymentB.

8.6 Documentum Server full-text objects and initialization files

HA requires more configuration of the full-text indexing system than a basic installation. The following illustration shows the relationships between the configuration objects for the full-text indexes, index servers, and index agents and the Documentum Server and xPlore initialization files.

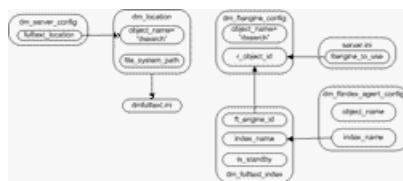


Figure 8-4: Full-text index objects

The following objects or properties are created in each repository:

- `dm_server_config.fulltext_location` property specifies the `dm_location` object (dsearch for xPlore).
- `dm_location` object (dsearch for xPlore). The property `file_system_path` specifies the path to `dmfulltext.ini`.
- `dmfulltext.ini` file is created when the server is installed. Contains information used by Documentum Server to find the index agent query plug-in binary files.

Location:

```
<DOCUMENTUM_HOME>/fulltext/dsearch/dmfulltext.ini
```

- `server.ini ftengine_to_use` parameter references the `dm_ftengine_config.r_object_id`. Location:

```
<DOCUMENTUM_HOME>/dba/config/<repository>/server.ini
```



Note: `ftengine_to_use` is only used in dual mode.

- `dm_fulltext_index.ft_engine_id` specifies the `dm_ftengine_config` object to use.

`dm_fulltext_index.is_standby` specifies whether the `dm_fulltext_index` object is on standby and the index to use for queries. If the `is_standby` value is 0, then the index specified by the `dm_fulltext_index` object is the default and is used for queries; if the value is 1, then the index specified by the `dm_fulltext_index` object is in standby mode and is not used for queries.



Notes

- The `ftengine_to_use` property in `server.ini` overrides the `dm_fulltext_index.is_standby` setting; that is, queries are sent to the index that is referenced by the `dm_fulltext_index` object, which is, in turn, specified by the `ftengine_to_use` property, even if the `dm_fulltext_index` object's `is_standby=1`.
- `dm_fulltext_index.is_standby` does not set the index to use for indexing requests. Both index engines service all indexing requests.
- `dm_ftindex_agent_config` represents an index agent configured for the repository. Its properties record status information and configuration information about the index agent.

Chapter 9

Upgrading xPlore

9.1 Upgrade overview

You can upgrade to the current version of xPlore from version 1.6 or a later release.

For non-HA configurations, the upgrade procedure is as follows:

- In-place upgrade

An in-place upgrade basically installs the target xPlore version into the installation directory of the existing instance.

Make sure that the operating system on which the existing xPlore instance installed is also supported by the target xPlore version.

If you upgrade xPlore in an HA configuration, you must perform some additional steps.

If an xPlore instance and a CPS instance reside on the same host, always upgrade the xPlore instance first.

9.2 Performing pre-upgrade tasks

Perform the following pre-upgrade tasks before upgrading xPlore:

- Deploy or upgrade to the relevant version of Documentum Server. Install the latest cumulative patch that is available for your version of Documentum Server at the time of installation. If you upgrade Documentum Server, restart Documentum Server before upgrading xPlore.



Note: If you do not install the latest Documentum Server patch that is available at the time of installation, xPlore may not work properly.

- Apply the latest available patch for multiple instance xPlore deployments. After applying the patch, start all instances and then stop them before upgrading. If the Documentum Server version is supported and if the Documentum Server worked with xPlore, and for single instance xPlore deployments, it is not a prerequisite to apply the latest patch.
- Back up your xPlore system using one of the following approaches based on your RTO (Recovery Time Objective). Backups are warm (search only) or cold (offline).
 - Volume-based backup: If you have a short RTO, perform a full backup of disk blocks using a third-party product such as EMC Timefinder. This backup method allows you to quickly restore your system in case of upgrade failure.

- Full file-based backup: If you have a long RTO, back up the complete `<xplore_home>` directory, including configuration, index data, and dblog files.



Note: Do not perform a backup until the final merge has moved your old Lucene index files into xDB.

For detailed information about backups, see the *Backup and Restore* chapter in the *OpenText Documentum xPlore Administration and Development Guide*.

- Make sure that no collections are off_line. An off_line collection is corrupted and you must restore it from a previous backup or remove it.



Note: Your xPlore system must meet this requirement before upgrade; otherwise, xPlore instance upgrade will fail and you will have to restore from your xPlore backup, fix this blocking issue, and start all over again.

To check whether there is any offline collection, open the file `XhiveDatabase.bootstrap` located in `<xplore_home>/config` and search for the string `usable="false"`. If the string cannot be found, it means there are no offline collections.

If an offline collection exists, restore it from a previous backup. For instructions on performing an offline restore, see the *Backup and Restore* chapter in the *OpenText Documentum xPlore Administration and Development Guide*.

If an offline collection exists with no previous backup available, perform the following steps to remove it:

1. Force-detach the offline collection using xPlore Administrator.
2. Shut down xPlore.
3. Remove all segments with the attribute `usable="false"` from the `XhiveDatabase.bootstrap`.
4. Remove the detached collection from `indexserverconfig.xml`.



Note: If you remove any offline collections, you must run the `ftintegrity` and `ac replication` scripts to refeed missing documents into the collection after system upgrade.

- Decide the xDB temp folder size and xDB transaction log size.

If you upgrade to xPlore 1.3 or later versions, you may have to modify the value of `xdb.lucene.temp.path` to make sure the xDB temp folder is large enough to hold the largest generated Index entry.

`xdb.lucene.temp.path` can be found in `xdb.properties`, which is located in the directory `WEB-INF/classes` of both primary and secondary instances.

Additionally, the folders where xDB transaction log resides, `xplore_home/config/wal` for the primary instance and `xplore_home/config/node_name_wal` for secondary instances, must have enough space to hold the largest index size because final merge may work under transaction logging mode.

9.3 Performing an in-place xPlore upgrade

To perform an in-place xPlore upgrade in a non-HA (high availability) environment:

1. Stop all xPlore instances by running the stop scripts in `<xplore_home>/<wildfly_version>/server`.

Make sure the xDB has been shut down cleanly. A clean shutdown allows the server to write all modified pages back to disk and the log files are not needed on startup.

Messages similar to the following in the dsearch log file indicate that the xPlore instance and xDB have been shut down cleanly.

```
2016-04-12 16:53:43,577 INFO [RMI TCP Connection(35)-10.32.122.149]
c.e.d.core.fulltext.indexserver.core.ESSNode - The xPlore instance
PrimaryDsearch shutdown complete.
```



Note: xDB must be shut down cleanly; otherwise, xPlore instance upgrade will fail and you will have to restore from your xPlore backup, fix this blocking issue, and start all over again.

2. Run `setup.exe` (Windows) or `setup.bin` (Linux) and choose the existing xPlore root directory as the installation directory. Installer detects your existing xPlore and asks whether you wish to upgrade. Choose **Yes** to update xPlore program files.



Note: If you do not install into the existing xPlore root directory, your existing installation is not upgraded.

3. Upgrade the primary xPlore instance.

- a. Start the xPlore configuration program by running `<xplore_home>/setup/dsearch\configDsearch.bat` (Windows) or `<xplore_home>/setup/dsearch/configDsearch.sh` (Linux).
- b. Choose **Upgrade Existing Server Instance** and then select the primary instance for upgrade.
- c. Enter the installation owner password and optional domain name.



Note: During the upgrade process, the configuration program first performs a pre-upgrade check to see if the xDB has been shut down cleanly and there are no off_line collections. If these pre-upgrade requirements are not met, the upgrade will fail and you will have to restore from your xPlore backup, fix these blocking issues, and start all over again.

- d. Start the primary instance. Keep all other instances stopped. Make sure that the latest patch has been installed before starting the primary instance.
4. Upgrade secondary xPlore instances. Repeat these steps for each secondary instance you want to upgrade.

- a. If the secondary instance resides on a different host than the primary instance, and you have not run xPlore Installer to update xPlore program files on this host, run `setup.exe` (Windows) or `setup.bin` (Linux) and choose the existing xPlore root directory as the installation directory. Installer detects your existing xPlore and asks whether you wish to upgrade. Choose **Yes** to update xPlore program files.

 **Note:** If you do not install into the existing xPlore root directory, your existing installation is not upgraded.

- b. Make sure the primary xPlore instance is running and stop all other secondary instances, if any.
- c. Choose **Upgrade Existing Server Instance** and then select the secondary instance for upgrade. Make sure that the latest patch has been installed before upgrading the instance.
- d. Enter the installation owner password and optional domain name.

 **Note:** During the upgrade process, the configuration program first performs a pre-upgrade check to see if the xDB has been shut down cleanly and there are no offline collections. If these pre-upgrade requirements are not met, the upgrade will fail and you will have to restore from your xPlore backup, fix these blocking issues, and start all over again.

- e. Start the secondary instance you just upgraded. Keep other secondary instances stopped.
5. Upgrade remote CPS instances, if any.
6. Upgrade is complete. You can review the installation log for any warnings or errors.

9.4 Performing post-upgrade tasks

After upgrade, perform the following steps:

1. Recreate index agents. See “[Configuring the index agent](#)” on page 45.

 **Note:** To ensure index agent refeed tasks are created, upload `ids.txt` and `ids.txt.old` from the `IndexAgent.war\WEB-INF\classes` folder of the old WildFly deployment directory.

2. If you upgraded from an xPlore deployment that was SSL-enabled, the SSL configurations are lost and you need to re-enable SSL in the new deployment. For instructions on enabling SSL in xPlore, see “[Configuring SSL connections](#)” on page 36
3. Test your upgraded xPlore installation. See “[Validating installation](#)” on page 28.
4. Back up the upgraded xPlore instances.



Note: You cannot restore an xPlore 1.0 or 1.1 backup because the xDB version has changed. Back up the upgraded instances after the final merge has moved your old Lucene index files into xDB.

9.5 Upgrading xPlore active-active HA

9.5.1 Performing an in-place xPlore HA upgrade

Use the following steps to perform an in-place xPlore upgrade in an active-active HA configuration.

1. On both the primary and secondary Documentum Server, set the `ftengine_to_use` value in `server.ini` to configure queries to be serviced by the standby xPlore instance. The `server.ini` file is located in `<DOCUMENTUM_HOME>/dba/config/<repository>`.
2. Upgrade the primary xPlore instance using the steps described in “[Performing an in-place xPlore upgrade](#)” on page 89. The search service still runs during the upgrade.
3. Shut down the standby xPlore instance and index agent.
4. Execute the `create_fulltext_objects_ha.ebs` script with the `HACleanupBeforeUpgradeStep`. This script deletes all `dm_ftengine_config` and `dm_fulltext_indexSyntax`:


```
dmbasic -f create_fulltext_objects_ha.ebs -e HACleanupBeforeUpgradeStep --<repository><superuser><password>
```
- objects.
5. Configure the index agent for the primary xPlore instance. See “[Configuring the index agent](#)” on page 45.
6. On both the primary and secondary Documentum Server, remove the `ftengine_to_use` setting in `server.ini`, and then restart the Documentum Servers.
7. Upgrade the standby xPlore instance using the steps described in “[Performing an in-place xPlore upgrade](#)” on page 89. Delete the index agent on the standby xPlore instance.
8. Configure the index agent for the primary and standby xPlore instances using the steps described in “[Configuring the index agent](#)” on page 45. For the standby xPlore instance, choose **Create a New Standby Indexagent**.
9. Restart the Documentum Servers.

Chapter 10

Migrating from FAST to xPlore

10.1 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.



Note: During migration, user and group permissions are not fully updated. Search may allow results that would not be returned when security has migrated. If this security is not acceptable, switch to Documentum Server security, restart Documentum Server, migrate, switch security, and then restart Documentum Server again. For instructions, see “Managing Security” in *OpenText Documentum xPlore Administration and Development Guide*.

1. Deploy or upgrade to Documentum Server 7.1 or later. Configure at least one Documentum repository and one connection broker. Configure at least one Documentum repository and one connection broker.
2. 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**.
3. Remove FAST: Uninstall the FAST indexing server using the instructions in your version. Delete the Documentum index agent for FAST and stop the Documentum Server. See “[Deleting an index agent](#)” on page 52.
4. Restart the Documentum Server. After successfully starting up Documentum Server and the connection broker, wait several minutes (typically 15 minutes) before installing the Documentum index agent.
5. Install the xPlore primary instance and start xPlore. See “[Installing the xPlore primary instance with Documentum Server 7.3 and later](#)” on page 26.
6. Install and configure a Documentum index agent for xPlore and start the index agent UI. See “[Configuring the index agent](#)” on page 45.
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 *OpenText Documentum xPlore Administration and Development Guide*.
7. 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.

After indexing completes, 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 *OpenText Documentum xPlore Administration and Development Guide*.

10.2 Migrating indexes (reindexing)

To migrate indexes from FAST to xPlore, reindex your content using an xPlore index agent. See “[Immediately replacing FAST with xPlore](#)” on page 93.

1. 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 *OpenText Documentum xPlore Administration and Development Guide*, Using the index agent filters section.
2. 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 *OpenText Documentum xPlore Administration and Development 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.
3. Temporarily configure multiple CPS daemons for reindexing.
 - a. Stop the CPS instance in xPlore administrator. **Choose Instances > Instance_name > Content Processing Service** and click **Stop CPS**.
 - b. Edit the CPS configuration file in the CPS host directory `<xPlore_home>/dsearch/cps/cps_daemon`.
 - c. Change the value of element `daemon_count` to 3 or more (default: 1).
 - d. Change the value of `connection_pool_size` to 2
 - e. Restart all xPlore instances.
 - f. Reset the CPS `daemon_count` to 1 and `connection_pool_size` to 5 after reindexing is complete.

10.3 Migrating large environments

10.3.1 Large migrations best practices

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.

Migration best practices:

- Plan indexing disk requirements with the xPlore sizing tool.
- Do full backups before migration and at points during migration. Reserve disk space for multiple backups.
- Test migration in a non-production environment.
- Change final merge interval. By default, changes are merged to the index every four hours to ensure low latency. Increase this interval during migration. See information about the setting *finalMergingInterval* in *xdb.properties*, in *OpenText Documentum xPlore Administration and Development Guide*.

10.3.2 Supporting a large number of ACLs

1. If your environment has more than 500,000 ACLs, turn off compression for ACLs.
2. Stop all xPlore instances.
3. Edit *indexserverconfig.xml*. Locate the element *dmftsecurity/acl_name* and *dmftsecurity/acl_domain*. Set the value of the *compress* attribute to *false*.
4. Save the config file and restart the primary instance, then restart all secondary instances.

10.3.3 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.

1. Create an xPlore secondary instance for each backfile collection you created.
2. Bind each backfile collection to the temporary xPlore instance.
3. Migrate data. See “[Migrating data to specific collections](#)” on page 96.
4. In addition to routing the requests to the backfile collection, make sure your routing class sets node hint to make those requests sent to the temporary xPlore instances which will use the temporary CPS instances.

5. Move backfile collections from the temporary instance to the primary instance. (Change binding using xPlore administrator.)
6. Delete the secondary (temporary) instances that you installed for migration.
7. If you add a routing class to route some type of requests to the temporary instance, remove that class after migration.

10.3.4 Ensuring CPU, disk, and I/O capacity

Indexing is both CPU and I/O intensive.

1. Temporarily increase CPU resources.
 - a. Bind extra collections to secondary instances, then move to primary instance when migration is finished.
 - b. Add more cores per CPU instance.
2. Improve disk I/O if CPU is sufficient but indexing still has a bottleneck.
 - a. See *OpenText Documentum xPlore Administration and Development Guide* for information on how to detect I/O bottlenecks.
 - b. Use striped disk allocations for xPlore data and log directories and for the index agent temporary staging area.

10.3.5 Migrating data to specific collections

By default, xPlore routes documents in a round-robin fashion over multiple collections. When you create multiple collections, xPlore can ingest documents simultaneously to different sets of Lucene indexes. You can target queries to the appropriate collection and speeds up response time.

1. Create at least three backfile collections using xPlore administrator and ingest documents to those collections. Add more collections if the hosts have good I/O capacity. If you use round-robin ingestion, you will not be able to perform bulk deletes of a collection from a specific file store.
2. Ingest data to backfile and day-forward collections using the index agent UI.
3. After data has been migrated, route new documents and updates to the day-forward collection.
4. Optional: Merge collections for recent data to the day-forward (ongoing) collection.
5. Start ingestion to the backfile collections using the index agent UI.
6. After data has been migrated, direct new data and updates to a day-forward collection.
7. Optional: Merge collections for recent data to day-forward (ongoing) collection.

You can determine a cutoff date for legacy data and then route your legacy data to backfile collections. Add routing logic to a custom routing class to support the cutoff date. Use 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 *OpenText Documentum xPlore Administration and 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. For a description of this routing customization, see “[Custom routing](#)” on page 98. For implementation details, see “[Custom routing class](#)” in *OpenText Documentum xPlore Administration and Development Guide*.

After ingestion, you can consolidate collections using xPlore administrator. You can also target queries to improve query performance and specificity. See “[Targeting queries to specific collections](#)” on page 98.

10.3.6 Migrating content based on document age

When you separate content into age-based collections, you can query recent data separately. Merges are minimized. To determine whether your query load is primarily for recent content, perform the following steps.

For large repositories with old data, separate old content from new content by collection. Old data is not queried very often, as shown in the diagram Time basis of queries:

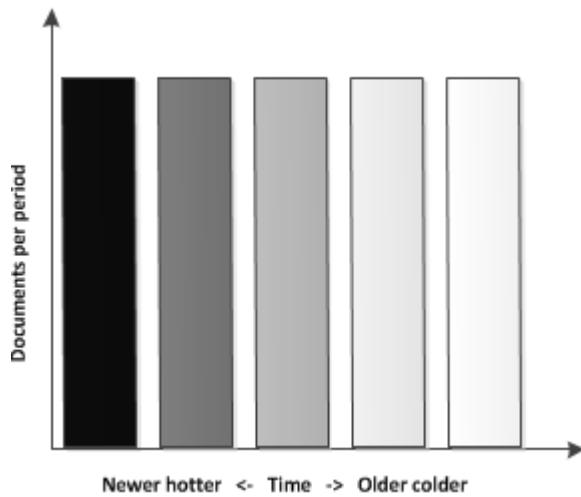


Figure 10-1: Time-based queries

1. Compare results from the DQL queries of your Documentum Server. First query:

```
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
```

Counts how many objects were modified and accessed within the past two years.

2. Second query:

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

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.

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. For a high-level description, see “[Custom routing](#)” on page 98.

10.3.7 Custom routing

The *OpenText Documentum xPlore Administration and Development Guide* fully describes routing customization. A custom routing class does the following:

1. Use logic to determine routing to a specific collection or set of collections.
2. Filter out content based on a date attribute.
3. Create and test the routing or filter class before you create the day-forward (ongoing) collection.

10.3.8 Targeting queries to specific collections

Default queries can be targeted to the most recent collection. Targeted queries can examine all collections in parallel.

Route queries with one of the following approaches:

- 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 *OpenText Documentum Search Development Guide*.

10.3.9 Tuning parameters for large migrations

Table 10-1: 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 <i>xplore_home/<wildfly_version>/server</i>	Required: At least 4 GB per xPlore instance.
xDB page server cache	WEB-INF/classes/indexserver-bootstrap.properties in the primary instance war file	Required: Stop all xPlore instances and increase <i>xhive-cache-pages</i> to 512 MB - 1 MB (in bytes). Restart xPlore instances.
Number of open files	<i>nofile</i> ulimit 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 <i>queue_size</i> in indexer to 1500.
Index agent callback queue size	indexagent.xml in index agent instance WEB-INF/classes	Required: Set <i>callback_queue_size</i> in 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.

10.4 Troubleshooting migration

If you installed Documentum Server 6.7 SPx and enabled FAST, you must manually install subscription query support for xPlore. See *OpenText Documentum xPlore Administration and Development Guide*.

Check the full-text config objects in the Documentum Server to make sure you have installed xPlore support. Then run fintegrity. Instructions are in *OpenText Documentum xPlore Administration and Development Guide*.

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.

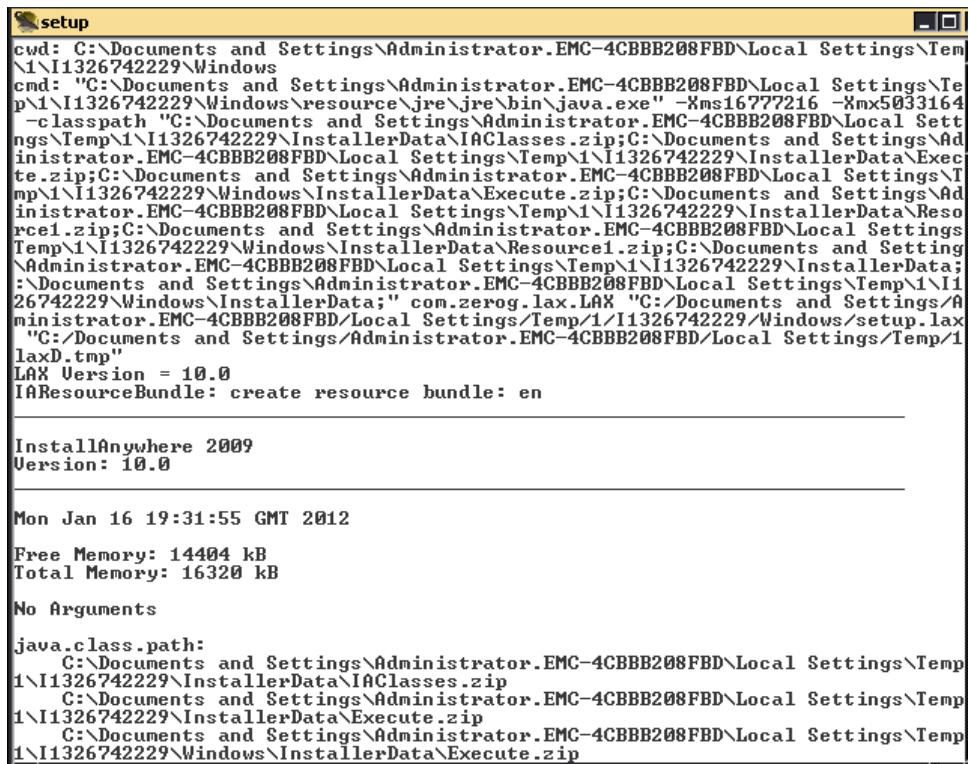
Chapter 11

Troubleshooting Installation

11.1 Debugging installation

To log debug output from the installer, run the installer using the following options:

- Windows: Hold down the control key while you launch the installer (setup.exe). Do not release the control key until the installer pops up a console window. The console window looks like the following:



```
setup
 cwd: C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\Windows
 cmd: "C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\Windows\resource\jre\jre\bin\java.exe" -Xms1677216 -Xmx5033164 -classpath "C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\IAClasses.zip;C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\Execute.zip;C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\Execute.zip;C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\Resource1.zip;C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\Windows\InstallerData\Resource1.zip;C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\Windows\InstallerData;" com.zerog.lax.LAX "C:/Documents and Settings\Administrator.EMC-4CBBB208FBD/Local Settings/Temp/1/I1326742229/Windows/setup.lax" "C:/Documents and Settings/Administrator.EMC-4CBBB208FBD/Local Settings/Temp/1/laxD.tmp"
 LAX Version = 10.0
 IAResourceBundle: create resource bundle: en

InstallAnywhere 2009
Version: 10.0

Mon Jan 16 19:31:55 GMT 2012
Free Memory: 14404 kB
Total Memory: 16320 kB
No Arguments
java.class.path:
  C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\IAClasses.zip
  C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\InstallerData\Execute.zip
  C:\Documents and Settings\Administrator.EMC-4CBBB208FBD\Local Settings\Temp\1\I1326742229\Windows\InstallerData\Execute.zip
```

Figure 11-1: Console Window

- Linux: Set LAX_DEBUG=true in the environment variables and then launch the installer. To redirect output from the console to a file, set LAX_DEBUG=file. After installation, a file named jx.log is generated in the install directory.

Checking installation versions

All xPlore instances and index agents should have the same installation version. You can check the version in the version.properties file in xplore_home/installinfo.

11.2 Fixing indexserverconfig upgrade issues

During xPlore upgrade, a script upgrades the main configuration file, indexserverconfig.xml, to support the new logging paradigm. If the script encounters errors, you must fix them before running xPlore. You can encounter the following errors:

- The file is updated. [error message and stack trace]. It's not a valid configuration file. The change is reverted.
- Failed to read the file (I/O exception).
- Failed to parse the file. (Not a valid XML file. Throws SAXException.)
- Failed to update the file [stack trace]. Other error, throws exception.

After you have fixed errors, you can run the script separately from the installer in order to upgrade your configuration. You can also run this script to upgrade a configuration file that is not in a 1.6 upgrade. The file will only run within a 1.6 xPlore installation.

1. Open a command-line window.
2. Change the working directory to the location of *xplore_home/dsearch/admin*.
3. Run the following CLI command with two parameters: The path to the script and the path to indexserverconfig.xml. Use the following syntax appropriate for your environment (Windows or Linux). The -f option is the path to the script followed by the script arguments. The command is case-insensitive on Windows.

```
xplore.bat -f scripts\UpdateConfigFile.groovy  
"C:/xPlore/config/indexserverconfig.xml"  
.xplore.sh -f ./scripts/UpdateConfigFile.groovy  
"/xPlore/config/indexserverconfig.xml"
```

If upgrade is successful, you see the following output:

```
The file is updated.  
It's a valid configuration file.
```

If the file was already updated, you see the following:

```
No need to update the config file.
```

For more information on using command-line arguments in xPlore, see the Automated Utilities (CLI) chapter in *OpenText Documentum xPlore Administration and Development Guide*.

11.3 Host name contains an underscore

The HTTP client specification does not permit underscores in URLs. Use the IP address or fully qualified domain name. The following example changes the host name to an IP address:

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,1,param_value[2]
SET>IP_address
save,c,1
```
4. Edit indexserverconfig.xml, which is located in *xplore_home/config*. Locate the node element for the primary node. Change the url attribute value to the IP address.
5. Save the configuration and restart xPlore instances.

11.4 Instance name fails

An instance name must begin with an alphabetic character followed by any combination of alphanumeric characters. If the name for an instance does not conform to this requirement, the configuration can fail with the error “failed to register *instance_name*”.

11.5 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.
- The instance configuration fails with the error “failed to register *instance_name*.” The name must begin with an alphabetic character followed by any combination of alphanumeric characters.

11.6 Cannot start 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

11.7 CPS instance does not start

If the CPS daemon has not shut down completely during xPlore upgrade, it is not replaced. The old CPS daemon runs, and the new one is incompletely installed. Follow these steps to fix it:

1. Turn off auto-restart of the Windows or Linux job.
2. Stop WildFly using the stop script.
3. Check the processes on the host to make sure the CPSDaemon process is not running. If it is running, kill the process and restart the host.
4. Run the xPlore configuration script to reconfigure a CPS instance.

11.8 xPlore administrator problems with Internet Explorer 8 and Internet Explorer 9 browsers

Modify Internet Explorer enhanced security in the browser to use xPlore administrator on Internet Explorer 8 and 9.

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 Internet Explorer and restart Internet Explorer.

11.9 Index agent errors

11.9.1 Global registry not available

Follow these troubleshooting steps in the following order.

1. Restart the global registry Documentum Server and connection broker.
2. Verify that the global registry repository projects to the same connection broker as the Documentum 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 Documentum 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 Documentum Server is not a global registry, designate it as one. Set the attribute *docbase_roles[0] = 'Global Registry'* in the *docbaseconfig* object.

11.9.2 Index agent configuration fails

Configuration fails when you run *configIndexagent.bat* or *configIndexagent.sh*.

1. Make sure that the index agent instance owner is the same as the xPlore instance owner.
2. Check the logs in *<xPlore_home>/setup/indexagent/logs* for errors.
3. Run the script again with a console. On Windows, control-click the batch file. You see a more detailed error description.
4. If the index agent is unable to create a domain, verify that you can ping the xPlore host from the index agent host. If not, add the xPlore host to the index agent hosts file.

11.9.3 Index agent version and OS

The index agent must be installed on the same OS as xPlore instances and have the same instance owner.

All xPlore instances and index agents should have the same installation version. You can check the version in the version.properties file in xplore_home/installinfo.

11.10 Linux errors

Failed to install WildFly

You see the following error:

```
ERROR [installer]
com.documentum.install.appserver.installanywhere.actions.InstallWildFlyAction -
Failed to install WildFly.
java.lang.Exception: No WildFly installer is available for your OS platform.
```

Workaround: Add the host to the hosts file in the etc directory.

Linux 'too many files open'

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

```
soft    nofile  1048576
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 12

Uninstalling xPlore

12.1 Uninstalling xPlore

To completely uninstall an xPlore installation, repeat the following procedure for every machine in your xPlore federation. Replace the variables index_agent, primary_name, and secondary_name with the names that you use when you configured index agent and xPlore instances.

1. Uninstall all index agents. See “[Deleting an index agent](#)” on page 52.
2. Delete all secondary xPlore and CPS instances using the xPlore configuration program on each instance: configDsearch.bat (Windows) or configDsearch.sh (Linux) in *xplore_home/setup/dsearch*. Choose **Delete Existing Server Instance**.
3. Delete the primary xPlore instance using the xPlore configuration program: configDsearch.bat (Windows) or configDsearch.sh (Linux) in *xplore_home/setup/dsearch*. Choose **Delete Existing Server Instance**.
4. Run the uninstallation program Uninstall.exe (Windows) or Uninstall (Linux) in the directory *<xplore_home>/Uninstall* and follow the instructions to complete the uninstallation process.

On Windows, if you ran the uninstallation program before deleting all xPlore and index agent instances, do the following:

1. Set the Windows services for the xPlore and index agent instances to manual.
2. Reboot the machine and delete *<xplore_home>*.

Chapter 13

References

13.1 xPlore ports

The following table describes the ports used by xPlore components.

Table 13-1: 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.

13.2 HTTP status codes

The following table describes the status codes that the URL `http://<hostname>:<port>/dsearch` returns:

Status code	Description
256	xPlore has been shut down with the URL <code>http://<hostname>:<portnumber>/dsearch/?action=shutdown</code>
257	xPlore is running and searchable
258	xPlore is running and indexable
259	xPlore is running, searchable and indexable

