

Oracle® Fusion Middleware

Autonomy Search Integration Sample Guide for Oracle
WebLogic Portal

10g Release 3 (10.3.5)

E15073-04

July 2012

Oracle Fusion Middleware Autonomy Search Integration Sample Guide for Oracle WebLogic Portal, 10g Release 3 (10.3.5)

E15073-04

Copyright © 2010, 2012 Oracle and/or its affiliates. All rights reserved.

Primary Author: William Witman

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	x
Conventions	x
 1 Introduction to Autonomy Search	
1.1 Search in the Portal Life Cycle	1-1
1.1.1 Architecture Phase	1-1
1.1.2 Development Phase	1-2
1.1.3 Staging Phase	1-2
1.1.4 Production Phase	1-2
1.2 Getting Started	1-2
1.2.1 Determining the Number of CPUs for Your Search Needs	1-4
1.2.2 Choosing an Operating System	1-4
1.2.3 System Requirements	1-4
1.2.4 Getting Support	1-5
1.3 Disabling Search/ Search Indexing for an Oracle WebLogic Portal Content Repository	1-5
 2 Requirements for Integrating Autonomy Search	
2.1 Autonomy Version Requirements	2-1
2.2 IDOL Server System Requirements	2-1
2.2.1 Supported Platforms	2-2
2.2.2 Minimum Hardware Requirements	2-2
2.2.3 Recommended Hardware Requirements	2-2
2.2.4 Additional Requirements	2-2
2.2.5 Additional Documentation	2-3
2.3 DiSH System Requirements	2-3
2.3.1 Additional Documentation	2-3
2.4 Autonomy Service Dashboard Requirements	2-4
2.4.1 Additional Documentation	2-4
2.5 Autonomy Portlet System Requirements	2-5
2.6 HTTP Fetch System Requirements	2-5
2.6.1 Additional Documentation	2-5

2.7	File System Fetch System Requirements	2-5
2.7.1	Additional Documentation	2-6
2.8	ODBC Fetch System Requirements	2-6
2.8.1	Additional Documentation	2-7
3	Configuring Your Portal to Use Autonomy Search	
3.1	Preparing to Develop a Portal with Autonomy Search.....	3-1
3.1.1	Configuring Autonomy Search Capabilities in Your Development Environment....	3-2
3.1.2	Disabling Oracle WebLogic Portal Content Management Full-Text Search.....	3-2
3.1.3	Disabling Autonomy's Services	3-2
3.1.4	Installing Autonomy Portlets.....	3-3
3.1.5	Enabling Autonomy Portlets in a Portal Application	3-7
3.2	Writing Autonomy-based Applications and Portlets.....	3-8
3.3	Creating Autonomy Search Portlets for WLP Content Repositories.....	3-8
3.4	Configuring Full-Text Wildcard Searches.....	3-8
4	Integrating Your Portal with Autonomy Search	
4.1	Adding the Autonomy Sample Implementation to the Portal Enterprise Application ...	4-2
4.2	Modifying the Oracle WebLogic Server Start Script to Start the Autonomy Search Engine	4-3
5	Architectural Considerations for Using Autonomy Search	
5.1	Understanding How Autonomy Search is Implemented	5-1
5.2	Deciding What Information to Index.....	5-2
5.2.1	Maximum Amount of Indexed Content.....	5-2
5.3	Architectural Recommendations	5-2
6	Configuring Multi-language Searching and Indexing	
6.1	One Language per Autonomy Server	6-1
6.2	One Language per Repository	6-2
6.3	Mixing Languages Within a Repository	6-2
6.3.1	Configuring Automatic Language Detection	6-3
6.3.2	Creating Queries	6-4
6.3.3	Query Text in Same Language and Any Encoding	6-4
6.3.4	Query Text in Same Language with Specific Encoding	6-4
6.3.5	Query Text in Another Language and Encoding.....	6-4
6.3.6	Query Across All Languages	6-4
6.3.7	Query Multiple Specific Languages.....	6-5
6.4	Enterprise Search for Microsoft Word, Excel, and PowerPoint Files in Multibyte Languages	6-5
6.4.1	Settings in omnislave.cfg	6-5
6.4.2	Settings in AutonomyIDOLServer.cfg.....	6-6
7	Staging Autonomy Search Functionality	
7.1	Installing Autonomy on Your Target Server	7-2

7.1.1	Supported Operating Systems	7-2
7.1.2	Installing Autonomy	7-2
7.2	Configuring Autonomy on Your Target Server	7-3
7.2.1	Configuring the Autonomy IDOL Server	7-3
7.2.2	Configuring the Autonomy DiSH	7-4
7.2.3	Configuring Agentstore	7-5
7.2.4	Configuring HTTP Fetch	7-5
7.2.5	Configuring File System Fetch	7-6
7.3	Setting Up Content Management Search	7-6
7.3.1	Configuring the Content Management Fetch	7-7
7.3.2	Configuring WLP Repositories for Full-Text Search	7-8
7.3.2.1	Configuring Oracle WebLogic Portal Repository Autonomy Sample Integration Properties	7-8
7.3.2.2	Editing Full-Text Search Properties	7-10
7.3.3	Troubleshooting Full-Text Search for WLP Content Management Repository	7-11
7.3.4	Configuring the WLP Content Management Fetch When Using a Non-WebLogic Portal Supported Operating System	7-12
7.4	Staging File System Fetch within a WebLogic Cluster	7-13
7.5	Starting the Autonomy Services	7-14
7.6	Stopping Autonomy Services in Windows 2000	7-17
7.7	Installing Autonomy Service Dashboard	7-17
7.7.1	Prepare the Dashboard for Installation	7-17
7.7.2	Deploy the Autonomy Service Dashboard	7-18

8 Managing Autonomy Search in Production

8.1	Using the Autonomy Service Dashboard	8-1
8.2	Re-Indexing WLP Repository Content	8-1

List of Examples

7-1	Example BEACMRepoFetch.cfg File.....	7-7
7-2	Example Autonomy Start Script.....	7-15
8-1	Example of Using the cm_indexer.properties File	8-2
8-2	Sample cm_indexer.properties File	8-2
8-3	Example of Using Command Line Arguments:	8-2

List of Figures

1-1	How Search Fits into the Four Phases of the Life Cycle.....	1-1
5-1	Diagram of an Oracle WebLogic Portal Integration with Autonomy Search Tools	5-2
7-1	Example of Oracle WebLogic Portal Cluster Using Autonomy.....	7-1
7-2	Example Remote Autonomy Installation on a non-Oracle WebLogic Portal supported operating system.....	7-12

List of Tables

1-1	Autonomy Search Components Optionally Used with Oracle WebLogic Portal	1-3
2-1	System Requirements for DiSH	2-3
2-2	System Requirements for HTTP Fetch.....	2-5
2-3	System Requirements for File System Fetch	2-6
2-4	System Requirements for ODBC Fetch.....	2-7
3-1	Advanced Repository Properties.....	3-2
3-2	Autonomy Portlets.....	3-3
3-3	Helpful Content Search APIs	3-8
7-1	Autonomy Components and Their Respective Configuration Files	7-3
7-2	Autonomy Sample Integration Properties for WLP Repositories	7-9
7-3	Required Settings for Full-Text Search	7-10
8-1	Command Line Arguments for the cm_index_data Script.....	8-2

Preface

This guide describes how to configure Oracle WebLogic Portal to use Autonomy search, integrate your portal with Autonomy search, and perform all aspects of developing and configuring portlets to take advantage of Autonomy's capabilities.

Note: For more information on configuring full text search, see "Adding Full Text Search Capabilities to Oracle WebLogic Portal" in the *Oracle Fusion Middleware Content Management Guide for Oracle WebLogic Portal*

You can also configure Oracle WebLogic Portal to perform full-text searches against content in an Oracle WebCenter Content-based repository. WebCenter Content repositories include built-in support for full text search in Oracle WebLogic Portal. For more information on Oracle Universal Content Management, see *Oracle WebLogic Portal UCM VCR Adapter Guide for Oracle WebLogic Portal*.

If you would like to use Autonomy Enterprise Search, you must first purchase the required license and obtain the binaries from Autonomy Corporation at <http://www.autonomy.com>.

Audience

This document is intended for developers and portal administrators.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Oracle Fusion Middleware Portal Development Guide for Oracle WebLogic Portal*
- *Oracle Fusion Middleware Content Management SPI Development Guide for Oracle WebLogic Portal*
- *Oracle Fusion Middleware Content Management Guide for Oracle WebLogic Portal*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction to Autonomy Search

This chapter includes the following sections:

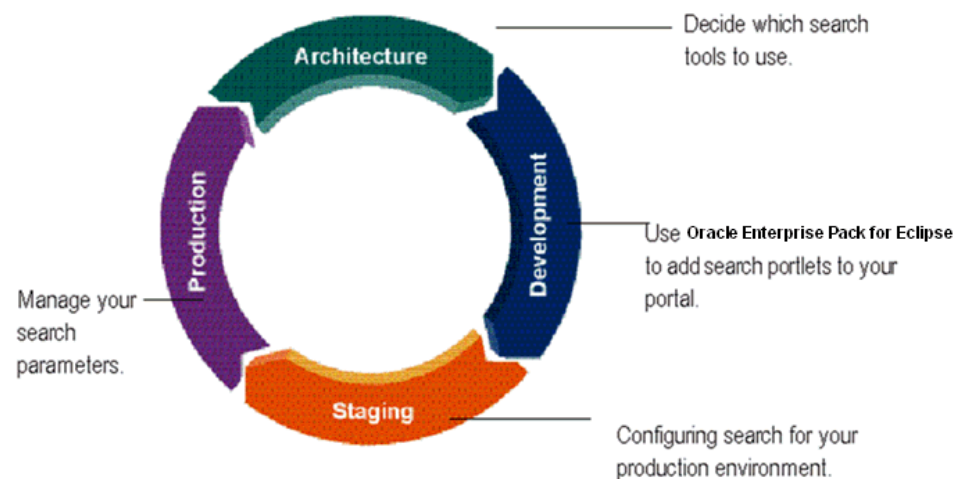
- [Section 1.1, "Search in the Portal Life Cycle"](#)
- [Section 1.2, "Getting Started"](#)
- [Section 1.3, "Disabling Search/ Search Indexing for an Oracle WebLogic Portal Content Repository"](#)

1.1 Search in the Portal Life Cycle

The tasks in this guide are organized according to the portal life cycle. For more information about the portal life cycle, see the *Oracle Fusion Middleware Overview for Oracle WebLogic Portal*. The portal life cycle contains four phases: architecture, development, staging, and production.

[Figure 1–1](#) shows how search fits into the portal life cycle.

Figure 1–1 How Search Fits into the Four Phases of the Life Cycle



1.1.1 Architecture Phase

During the architecture phase, you determine what enterprise content you want to make available for your portal and who within your portal environment will be able to search this information.

The following chapter describes tasks within the architecture phase:

- [Chapter 5, "Architectural Considerations for Using Autonomy Search"](#)

1.1.2 Development Phase

During the development phase, you add search portlets to your portal, use the APIs to retrieve content, and optionally, write portlets to surface search features for your portal users.

The following chapter describes tasks within the development phase:

- [Chapter 3, "Configuring Your Portal to Use Autonomy Search"](#)
- [Chapter 6, "Configuring Multi-language Searching and Indexing"](#)

1.1.3 Staging Phase

The staging phase is when you prepare your production environment. During this phase, you reconfigure your search configuration to match your deployment configuration and enable tools to configure search when running in a production environment.

The following chapter describes tasks within the staging phase:

- [Chapter 7, "Staging Autonomy Search Functionality"](#)

1.1.4 Production Phase

After you deploy your application and are running in a production environment, you can adjust how your portal searches for content, including caches and search frequencies.

The following chapter describes tasks within the production phase:

- [Chapter 8, "Managing Autonomy Search in Production"](#)

1.2 Getting Started

Optional search features provided by Autonomy include the following:

- Natural language queries can be input to the IDOL Server and will be processed based on the words contained in the queries.
- Basic XML Search - basic features to index and search XML documents Natural Language Support.
- Relevance Ranking. Each retrieval operation produces a relevancy score which can be used in the search results interface
- Document Similarity Search – "More Like This" feature using keyword similarity between documents.
- Proximity Controls. Basic Boolean, proximity, and field searches are provided.

[Table 1–1](#) lists the components of Autonomy search tools and what each provides.

Table 1–1 Autonomy Search Components Optionally Used with Oracle WebLogic Portal

Autonomy Component	What It Does:
Autonomy IDOL Server	<p>The Intelligent Data Operating Layer (IDOL) server is responsible for indexing content as well as processing content queries made from your portal.</p> <p>For more information about the Autonomy IDOL Server, see the Autonomy IDOL Server documentation.</p>
Autonomy DiSH	<p>The Distributed Service Handler – DiSH, provides the crucial maintenance, administration, control and monitoring functionality of the Intelligent Data Operating Layer (IDOL). DiSH delivers a unified way to communicate with all Autonomy services from a centralized location.</p> <p>DiSH can be managed with the Autonomy Service Dashboard. For more information about the Autonomy DiSH, see the Autonomy DiSH documentation.</p>
Autonomy Service Dashboard	<p>The Autonomy Service Dashboard is a standalone front-end web application that communicates with one or more Autonomy Distributed Service Handler (DiSH) modules that provide the back-end process for monitoring and controlling all the Autonomy child services, such as fetches.</p> <p>For more information about the Autonomy Service Dashboard, see the Autonomy DiSH documentation.</p>
Autonomy HTTP Fetch	<p>HTTP Fetch allows documents from internet or intranet sites to be aggregated from remote servers and indexed into Autonomy IDOL server.</p> <p>For more information about the HTTPFetch, see the Autonomy HTTP Fetch documentation.</p>
Autonomy ODBC Fetch	<p>ODBC Fetch is an Autonomy connector that automatically retrieves data that is stored in ODBC data sources, imports it into IDX file format and indexes it into Autonomy IDOL server.</p> <p>For more information about the ODBC Fetch, see the Autonomy ODBC Fetch documentation.</p>
Autonomy File System Fetch	<p>File System Fetch analyzes file systems on local or network machines (including Novell, NT, UNIX file systems and Samba-mounted servers) for new documents to aggregate into the Autonomy IDOL server. It keeps the IDOL server's view of the file system in sync so that files deleted are automatically removed from IDOL server, and modifications to files are reflected automatically.</p> <p>Note: If a file name contains any Japanese characters using Shift JIS encoding, Autonomy will not index them. This means that if a file with Shift JIS characters in the file name are placed in a directory to be indexed by the File System Fetch utility, it will not be indexed by Autonomy and not be returned within the search results provided by the Enterprise Search portlet. Therefore, you must rename any files that contain Shift JIS characters to a name without any Shift JIS characters.</p> <p>For more information about the File System Fetch, see the Autonomy File System Fetch documentation.</p>

Table 1–1 (Cont.) Autonomy Search Components Optionally Used with Oracle WebLogic

Autonomy Component	What It Does:
Autonomy Portlets	<p>Autonomy portlets are designed to integrate search functionality with your portal.</p> <p>For more information about the Autonomy portlets, see Section 3.1.4, "Installing Autonomy Portlets".</p> <p>For additional documentation on the Autonomy portlets, see the <i>Autonomy Portlets for WebLogic Guide</i> or the <i>Autonomy Portlets User Guide</i>, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of these guides.</p>

1.2.1 Determining the Number of CPUs for Your Search Needs

The number of CPUs that you need for a production instance varies with the number and type of documents you are exposing, as well as the way they are exposed (for example, automated searching, user driven, and so on).

A single instance of one CPU can potentially support tens of thousands of users and millions of documents.

1.2.2 Choosing an Operating System

When you deploy your portal and install Autonomy within your portal environment, you will need to install the operating system-specific version of Autonomy on your server on which you run the Autonomy services. For more information about installing and deploying the Autonomy services, see [Chapter 7, "Staging Autonomy Search Functionality."](#)

Note: Autonomy binary executable files are named with a .exe extension (Windows style) for all operating systems.

Additionally, note the following:

- If you upgraded to Oracle WebLogic Portal 10.3.4, during development mode Autonomy services are automatically started for the operating system of the host computer. This allows developers to use Autonomy during portal development.
- If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately, you must modify your Oracle WebLogic Server start script to start the Autonomy search engine for Autonomy services to automatically start the operating system of the host computer. For details on modifying the Oracle WebLogic Server start script to start the Autonomy search engine, see [Chapter 4.2, "Modifying the Oracle WebLogic Server Start Script to Start the Autonomy Search Engine."](#)

1.2.3 System Requirements

When configuring Autonomy search for your portal application, please note Autonomy's system requirements, see the Autonomy documentation.

1.2.4 Getting Support

Oracle Support Services does not provide support for Autonomy search. When you pay for your Autonomy license, Autonomy Corporation should provide you access to the Autonomy Customer Support Center for your Autonomy support needs.

1.3 Disabling Search/ Search Indexing for an Oracle WebLogic Portal Content Repository

To disable search indexing:

1. Start your portal domain.
2. Start the Oracle WebLogic Portal Administration Console.
3. Select **Content > Content Management** from the navigation menu at the top of the console.
4. Select **Manage | Repositories**.
5. In the resource tree, click the repository for which you want to disable search indexing.
6. In the Summary tab, click **Advanced** to view the Edit Advanced Properties for Repository dialog.
7. In the Edit Advanced Properties for Repository dialog, clear the **Search Enabled/Search Indexing Enabled** check boxes.
8. When finished making changes, click **Save**.

Requirements for Integrating Autonomy Search

This chapter discusses Autonomy version requirements and system requirements for integrating Autonomy search with Oracle WebLogic Portal.

When considering requirements for using Autonomy search, you should first refer to the Oracle WebLogic Server system requirements, see "Oracle Fusion Middleware Supported System Configurations."

Autonomy's system requirements are a subset of the Oracle WebLogic Server requirements.

This chapter includes the following sections:

- [Section 2.1, "Autonomy Version Requirements"](#)
- [Section 2.2, "IDOL Server System Requirements"](#)
- [Section 2.3, "DiSH System Requirements"](#)
- [Section 2.4, "Autonomy Service Dashboard Requirements"](#)
- [Section 2.5, "Autonomy Portlet System Requirements"](#)
- [Section 2.6, "HTTP Fetch System Requirements"](#)
- [Section 2.7, "File System Fetch System Requirements"](#)
- [Section 2.8, "ODBC Fetch System Requirements"](#)

2.1 Autonomy Version Requirements

To integrate Oracle WebLogic Portal with Autonomy, you must add the Autonomy sample search implementation—created by Oracle software developers—to the portal enterprise application. The Autonomy sample search implementation is only compatible with Autonomy version 4.6.0. For this reason, you should use only Autonomy 4.6.0 with Oracle WebLogic Portal. Note that you must obtain the binaries from Autonomy Corporation at <http://www.autonomy.com> to use in conjunction with the sample Autonomy integration code.

2.2 IDOL Server System Requirements

Autonomy's Intelligent Data Operating Layer (IDOL) server is the main component that is required to use Autonomy search capabilities with Oracle WebLogic Portal. The IDOL server should be installed by a system administrator.

2.2.1 Supported Platforms

- Microsoft Windows NT4, 2000, XP and 2003
- Linux (all versions) kernel 2.2, 2.4 and 2.6
- Sun Solaris for SPARC versions 5 - 9
- Sun Solaris for Intel version 9
- AIX version 4.3, 5 and 5.1
- HP-UX for PA-RISC version 10, 11 and 11i
- HP-UX for Itanium version 11i
- Tru64 version 5.1
- IDOL server also supports other POSIX UNIX versions on request.

Note: Autonomy search engine and fetch processes might not start properly on Unix platforms due to missing linked libraries.

Autonomy executable files and shared libraries require certain system libraries to be installed and accessible, via the library path, in order to start properly. If your target server does not have these libraries, one or more of the following processes might not start:

AutonomyDiSH.exe, AutonomyIDOLServer.exe, content.exe, category.exe, community.exe, agenstore.exe, BEACMRepoFetch.exe, FileSystemFetch.exe, HTTPFetch.exe.

For the processes which failed to start, validate the linked libraries by using the unix ldd command. If any issues are found, obtain and install the missing library file(s).

2.2.2 Minimum Hardware Requirements

- 1 Gb RAM
- 30 Gb Disk
- 1.5 GHz CPU

2.2.3 Recommended Hardware Requirements

- A dedicated SCSI disk
- 4 Gb RAM
- 100 Gb Disk
- A minimum of 2 dedicated CPU - XEON 3 GHz or above

2.2.4 Additional Requirements

- You cannot run IDOL server with restricted file system permissions (for example disk quotas, file handle limits or memory limits).
- Your file system must permit file locking (this means that you cannot run IDOL server on an NFS mount, for example).
- Your network must support TCP/IP.

- If you upgraded to Oracle WebLogic Portal 10.3.4, the Autonomy license that is delivered with Oracle WebLogic Portal allows you to index 500,000 pieces of content. A single IDOL server can hold an approximate maximum of 8 million document sections (depending on the functionality and performance required). However, the license included with previous versions of Oracle WebLogic Portal allows only 500,000. The current version of Oracle WebLogic Portal does not include an Autonomy license.
- If you are running anti-virus software on the machine that hosts IDOL server, you should ensure that it does not monitor the IDOL server directories as this can have a serious impact on IDOL server's performance.

2.2.5 Additional Documentation

For more information about using the Autonomy IDOL server, see the *IDOL Server Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

For more information about configuring the IDOL server for use in your portal environment, see "Oracle Fusion Middleware Supported System Configurations."

2.3 DiSH System Requirements

The Autonomy DiSH (Distributed Service Handler) provides a central point from which you can manage Autonomy applications and/or portlets. [Table 2-1](#) lists the system requirements for the Autonomy DiSH.

2.3.1 Additional Documentation

For more information about using the Autonomy DiSH and the Autonomy Service Dashboard which is used to monitor the DiSH, see the *Autonomy Distributed Service Handler (DiSH) Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

For more information about configuring DiSH for use in your portal environment, see [Section 7.2.2, "Configuring the Autonomy DiSH."](#)

Table 2-1 System Requirements for DiSH

Supported Platform	Minimum Server Requirements
Microsoft Windows 98, NT 4, and 2000	<ul style="list-style-type: none"> ■ 200 MHz Pentium processor ■ 64 MB RAM ■ 1 GB hard disk recommended
Unix	<ul style="list-style-type: none"> ■ 128 MB of RAM ■ 1 GB hard disk recommended
Solaris	<ul style="list-style-type: none"> ■ 128 MB of RAM ■ 1 GB hard disk recommended

Note: Autonomy search engine and fetch processes might not start properly on Unix platforms due to missing linked libraries.

Autonomy executable files and shared libraries require certain system libraries to be installed and accessible, via the library path, in order to start properly. If your target server does not have these libraries, one or more of the following processes might not start:

AutonomyDiSH.exe, AutonomyIDOLServer.exe, content.exe, category.exe, community.exe, agenstore.exe, BEACMRRepoFetch.exe, FileSystemFetch.exe, HTTPFetch.exe.

For the processes which failed to start, validate the linked libraries by using the unix ldd command. If any issues are found, obtain and install the missing library file(s).

2.4 Autonomy Service Dashboard Requirements

To run Autonomy Service Dashboard you need a third party application server that is J2EE 1.3 compliant. You also need a Java Development Kit (JDK) installation, version 1.4.

Note: If you are using a web server that is separate from your application server, you must ensure that they are configured to run together.

Autonomy Service Dashboard runs with the following application servers:

- JRun 4.0 or higher
- Tomcat 4.0.0 or higher
- Resin 2.1.1 or higher
- ServletExec 4.2 or higher
- WebLogic 6.1 or higher
- WebSphere 5

These application servers all include an internal web server component that Autonomy Service Dashboard can operate with. Alternatively, you can integrate your application server with an external web server such as:

- Microsoft IIS version 5 and higher
- Apache 2 version 5 and higher

2.4.1 Additional Documentation

For additional information about using HTTP Fetch, see the *Autonomy DiSH Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

For additional information about installing the Autonomy Service Dashboard for use within your portal environment, see the [Section 7.7, "Installing Autonomy Service Dashboard,"](#).

2.5 Autonomy Portlet System Requirements

Autonomy portlets require the same system requirements as the application server from which they are run. For additional information, see "Oracle Fusion Middleware Supported System Configurations."

2.6 HTTP Fetch System Requirements

HTTP Fetch is a connector that is used to index content from web sites. HTTP Fetch should be installed by the system administrator as part of a larger Autonomy system (a system that includes Autonomy IDOL server and an interface for the information stored in IDOL server). [Table 2-2](#) lists the system requirements for HTTP Fetch.

2.6.1 Additional Documentation

For additional information about using HTTP Fetch, see the *Autonomy HTTP Fetch Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

For additional information about configuring HTTP Fetch for use within your portal environment, see [Section 7.2.4, "Configuring HTTP Fetch."](#)

Table 2-2 System Requirements for HTTP Fetch

Supported Platform	Minimum Server Requirements
Microsoft Windows NT and 2000 (Intel)	<ul style="list-style-type: none"> 200 MHz Pentium processor 64 MB RAM 2 GB hard disk recommended
Unix	<ul style="list-style-type: none"> 128 MB of RAM 2 GB hard disk recommended 2.2.12 kernel
Solaris 2.5 (Sun Sparc)	<ul style="list-style-type: none"> 128 MB of RAM 2 GB hard disk recommended

Note: Autonomy search engine and fetch processes might not start properly on Unix platforms due to missing linked libraries.

Autonomy executable files and shared libraries require certain system libraries to be installed and accessible, via the library path, in order to start properly. If your target server does not have these libraries, one or more of the following processes might not start:

AutonomyDiSH.exe, AutonomyIDOLServer.exe, content.exe, category.exe, community.exe, agenstore.exe, BEACMRRepoFetch.exe, FileSystemFetch.exe, HTTPFetch.exe.

For the processes which failed to start, validate the linked libraries by using the unix ldd command. If any issues are found, obtain and install the missing library file(s).

2.7 File System Fetch System Requirements

File System Fetch should be installed by the system administrator as part of a larger Autonomy system (that is a system that includes Autonomy IDOL server and an

interface for the information stored in IDOL server). [Table 2–3](#) lists the system requirements for File System Fetch.

2.7.1 Additional Documentation

For additional information about using File System Fetch, see the *Autonomy File System Fetch Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

For additional information about configuring File System Fetch for use within your portal environment, see [Section 7.2.5, "Configuring File System Fetch."](#)

Table 2–3 System Requirements for File System Fetch

Supported Platform	Minimum Server Requirements
Microsoft Windows NT 4, 2000 and XP	<ul style="list-style-type: none">■ 200 MHz Pentium processor■ 128 MB RAM■ 200 MB hard disk
Solaris	<ul style="list-style-type: none">■ 128 MB of RAM■ 2 MB hard disk
Linux	<ul style="list-style-type: none">■ 128 MB of RAM■ 2 GB hard disk

Note: This specification is dependent on the amount of data to be fetched. Due to substantially different disk usage patterns. File System Fetch also supports other POSIX UNIX versions on request.

Note: Autonomy search engine and fetch processes might not start properly on Unix platforms due to missing linked libraries.

Autonomy executable files and shared libraries require certain system libraries to be installed and accessible, via the library path, in order to start properly. If your target server does not have these libraries, one or more of the following processes might not start:

AutonomyDiSH.exe, AutonomyIDOLServer.exe, content.exe, category.exe, community.exe, agenstore.exe, BEACMRepoFetch.exe, FileSystemFetch.exe, HTTPFetch.exe.

For the processes which failed to start, validate the linked libraries by using the unix ldd command. If any issues are found, obtain and install the missing library file(s).

2.8 ODBC Fetch System Requirements

ODBC Fetch is used to index content from ODBC-compatible databases. ODBC Fetch should be installed by the system administrator as part of a larger Autonomy system.

[Table 2–4](#) lists the system requirements for ODBC Fetch.

2.8.1 Additional Documentation

For additional information about using ODBC Fetch, see the *Autonomy ODBC Fetch Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

Table 2–4 System Requirements for ODBC Fetch

Supported Platform	Minimum Server Requirements
Microsoft Windows NT 4 and 2000	<ul style="list-style-type: none"> ■ 200 MHz Pentium processor ■ 64 MB RAM ■ 2 GB hard disk
Solaris	<ul style="list-style-type: none"> ■ 128 MB of RAM ■ 2 GB hard disk
Linux	<ul style="list-style-type: none"> ■ 128 MB of RAM ■ 2 GB hard disk

- ODBC Fetch also supports other POSIX UNIX versions on request.
- This specification is dependent on the amount of data to be fetched. Due to substantially different disk usage patterns it is beneficial to run fetch and DRE processes on separate drives or partitions.
- You must have ODBC version 2.0 (or higher) drivers for the target database type(s). The Server also needs to have an appropriate Data source set up for each database containing content. This is done via the Control Panel. Data sources should be configured as System-wide (the "System DSN" tab) to ensure that they are accessible to all users.
- The ODBC Fetch requires an underlying ODBC Driver in order to connect to the data source. For example, if you want to run the ODBC Fetch on Solaris to access an Oracle 8.0 server running on the same or another platform, you must use an Oracle 8.0 compatible ODBC Driver for Solaris.

Configuring Your Portal to Use Autonomy Search

During the development phase, you can add Autonomy portlets to allow your portal users access to Autonomy search functions. You can also build your own portlets or customize existing ones.

This chapter discusses the following topics:

- [Section 3.1, "Preparing to Develop a Portal with Autonomy Search"](#)
- [Section 3.2, "Writing Autonomy-based Applications and Portlets"](#)
- [Section 3.3, "Creating Autonomy Search Portlets for WLP Content Repositories"](#)
- [Section 3.4, "Configuring Full-Text Wildcard Searches"](#)

Note: This note applies to you only if you are using Autonomy to search your Oracle WebLogic Portal Content Repository. Additionally, you do not need to configure your application to use the Autonomy integration code if you are only using Autonomy portlets.

After configuring your portal to use Autonomy to search, you must configure your application to use the sample Autonomy integration code provided with the Oracle WebLogic Portal. For details, see [Chapter 4, "Integrating Your Portal with Autonomy Search."](#)

3.1 Preparing to Develop a Portal with Autonomy Search

Before you add and create Autonomy search capabilities within your portal, you need to ensure your development environment has the necessary tools and the proper settings. Specifically, you need to add the Autonomy portlets to your web application library, and ensure that Autonomy search features are optimized for your environment.

This section includes the following topics:

- [Section 3.1.1, "Configuring Autonomy Search Capabilities in Your Development Environment"](#)
- [Section 3.1.2, "Disabling Oracle WebLogic Portal Content Management Full-Text Search"](#)
- [Section 3.1.3, "Disabling Autonomy's Services"](#)
- [Section 3.1.4, "Installing Autonomy Portlets"](#)
- [Section 3.1.5, "Enabling Autonomy Portlets in a Portal Application"](#)

3.1.1 Configuring Autonomy Search Capabilities in Your Development Environment

Running Autonomy's search services and Oracle WebLogic Portal full-text search while developing your portal could use unnecessary memory resources. If you do not need to run search capabilities continuously in your development environment, it is recommended that you configure a staging environment for your Autonomy search features, see [Chapter 7, "Staging Autonomy Search Functionality."](#)

Optionally, you could also disable search within your development environment. Disabling search involves disabling Oracle WebLogic Portal Content Management full-text search as well as turning off Autonomy's search services.

3.1.2 Disabling Oracle WebLogic Portal Content Management Full-Text Search

To disable Oracle WebLogic Portal Content Management full-text search:

1. Start your portal domain.
2. Start the Oracle WebLogic Portal Administration Console.
3. Select **Content** > **Content Management** from the navigation menu at the top of the console.
4. Select **Manage** | **Repositories**.
5. In the resource tree, click the repository for which you want to disable full-text search.
6. In the Advanced section, click **Advanced** to view the Edit Advanced Properties for Repository dialog.
7. In the Edit Advanced Properties for Repository dialog, edit the properties listed in [Table 3–1](#).

Table 3–1 *Advanced Repository Properties*

Advanced Property	What it does:
Search Indexing Enabled	Allows content for this repository to be indexed for portal full-text search against a repository which supports full-text search. This enables portal developers to use full-text content search in any portlets that they develop.
Full-Text Search Enabled	Enables users to search the repository using the full-text of the content.

1. When finished making changes, click **Save**.

Your modifications display in the Advanced section of the Summary page.

Note: After you make any changes to repository properties, Portal Administration Console users must log out and log back in to view the changes.

3.1.3 Disabling Autonomy's Services

If full-text search is not enabled on any Oracle WebLogic Portal repository, you can disable Autonomy's search services. When you disable Autonomy's services, the following features will not be available:

- Full-Text Search on a WLP repository

- Autonomy portlets
- Any application/portlet which uses the Autonomy API

To disable Autonomy's services:

1. Stop your portal domain.
2. Set the following environment variable on your portal domain server:
CONTENT_SEARCH_OPTION=none
3. Restart your portal domain.

3.1.4 Installing Autonomy Portlets

In order to use Autonomy's portlets within your portal application, you must first install and configure them.

For additional information about configuring these portlets, see the *Autonomy Portlets for WebLogic Administration Guide* or the *Autonomy Portlets User Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

The portlets are listed in [Table 3–2](#).

Table 3–2 Autonomy Portlets

Portlet	What It Provides:
Autonomy2DMap	The 2D Cluster Map is used to identify conceptual similarities and differences between clusters. Also based on JSP, the landscape is generated from the inter-relationships between clusters and the documents contained within those clusters. Designed to provide a single overview of the clusters contained within the data, clusters that are close together correlate to higher degrees of similarity, whilst dissimilar clusters are situated further apart. By scrolling over the ClusterMap automatic titles are generated and assigned to every cluster. By clicking on the cluster, the results and respective information can be viewed.
Agent	<p>The Agent portlet allows individual users to create their own personalized information channels, either from Natural Language, legacy Keywords, Boolean expressions, Parametric Searches or even simply by example. These agents then monitor all incoming information and can target and alert useful content on a continual basis, automatically.</p> <p>Users begin by customizing the Agent Portlet by setting up 'interest agents'. This is done by the user describing in plain natural language what it is they are interested in. The Interest Agent persistently identifies all relevant content and presents it in a concise personalized page, complete with URL links. As new information becomes available the agent will monitor new data submissions ensuring that the Portlet user is always provided with up-to-date information. Moreover, as user interests change users are also given the option to refine their interest by retraining the agents.</p>
Breaking News	The Breaking News portlet identifies what's new in the information space. Taking the cluster analysis from a previous time period and comparing it to a current one allows automatic identification of new clusters that weren't previously present, allowing automation of 'breaking news' pages, alerting to new areas of information or new interest trends in subscriber groups.
Community	The Community portlet notifies individual users of any agents that people in the work community may have set up using the Agent portlet, which resemble their own personalized agents. This Portlet brings together the benefits of collaboration, reducing duplicated effort as well as identifying experts within the organization.

Table 3–2 (Cont.) Autonomy Portlets

Portlet	What It Provides:
Cluster	<p>Cluster portlets provide a range of classification Portlets and visualization tools that can be added to the Oracle environment, further enriching the portal experience.</p> <p>Autonomy's automatic clustering features identify areas of intense research, breaking news or emerging trends and market opportunities based on information found within the knowledge base. Autonomy's Cluster Portlets can take large sets of document data or user-profile information and automatically identify the main set of concepts/ themes inherent within the knowledge base.</p> <p>Furthermore, clustering can be used in identifying the 'gap' between the users interests and the data being provided to the users thereby allowing 'knowledge/ content gaps' to be eliminated through provision or aggregation of further content relevant to the community.</p>
Expertise Locator	The Expertise Locator portlet allows users to find people who have been dealing with a specific subject by entering a brief natural language description of the subject. It returns all agents and profiles that match this description together with the names of the users who own the agents or profiles.
Hot News	The Hot News Cluster portlet can identify what is most popular or the main topics/clusters of information or interests found within the information assets or an organization. This allows the business to instantly receive a high-level view of the entire knowledge base providing a catalyst that enables informed decisions to be made faster.
Administration	The Administration portlet that enables you to administer and maintain all Autonomy Portlet settings from a central location.
Profile	The Profile portlet brings new documents to users attention based on each users individual interest and according to their profile. This Portlet creates a profile on each user based on the concepts of the documents that the user has been reading within the Autonomy Portlet suite (Agents, Retrieval, Clustering, Community and so on). Every time a user opens a new document within the Autonomy Portlet suite, the user's profile will be update based on the information read.
Retrieval	Autonomy's Retrieval portlet provides a fully-automated and precise means of retrieving information. It allows content to be searched in any language and any format, wherever it is stored, and presented with hyperlinks to similar information, automatically and in real-time. Unlike ordinary searches that look for keywords the Autonomy Retrieval Portlet allows you to enter a natural language query. The Retrieval Portlet submits the natural language query to one or more databases that have been set up, in order to find documents that are related to your query.
Similar People	The Similar People portlet notifies a user or other people in the same organization of other users who have been using the same type of documents that you have been looking at. This feature helps users avoid spending time on searching for information that may already be available.

Table 3–2 (Cont.) Autonomy Portlets

Portlet	What It Provides:
Spectrograph	<p>The Spectrograph portlet displays the relationship between clusters on successive periods and sets of data. Clusters are presented as a JSP-based spectrograph, whereby the x-axis represents information over time (enabling users to visualize how clusters develop over a given time period), whilst the y-axis represents the range of concepts defined within the knowledge base.</p> <p>Moreover, the spectrograph is able to display hot and breaking news in the same instance. The importance of clusters over time can be seen through the change of color and width. The color/ intensity of the lines is an indication of the size of cluster. The brighter colors indicate what is popular, and the width of the lines is an indication of the quality of the cluster. Navigation features are identical to the 2D ClusterMap enabling users to browse clusters with a click of the mouse.</p>

After installing the Autonomy portlets, you can use them or add them to your portal without customizing them. If installing the Autonomy portlets in a portal application, you should also configure an additional servlet, see [Section 3.1.5, "Enabling Autonomy Portlets in a Portal Application."](#)

For more information about working with portlets, see *Oracle Fusion Middleware Portlet Development Guide for Oracle WebLogic Portal*.

To add these portlets:

Note: These instructions assume you have already created and deployed your portal application.

1. Locate the `AutonomyPortlets.zip` file in your Oracle WebLogic Portal installation directory. For example,
`<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/common/AutonomyPortlets.zip`

Note: The path in the example is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the example. For this reason, you should change the path in your environment accordingly.

2. Extract the contents of the `AutonomyPortlets.zip` file to a temporary directory of your choosing. For example, `c:/temp/`.
3. Add the following to Oracle Enterprise Pack for Eclipse (OEPE) by right-clicking the appropriate folder and then selecting **Paste**:
 1. Copy the `autonomyPiB.jar` file to your `EAR/EARContent/APP-INF/lib` directory.
 2. Copy the portlet folder to your `webApp/webContent` directory. For example, `<your_projects>/w4WP_workspaces/<project_name>/<yourWebApp>/WebContent/`

3. Copy the `portalInabox.css` file to the `<yourWebApp>/webContent/portlets` directory that you added in the previous step.
4. Copy the `AutonomyPortletSettings.usr` file to the `userprofiles` subdirectory of your Datasync project directory. For example, `<yourDataSyncProject>/src/userprofiles/`.
5. Copy the `*.properties` files in the temporary `WEB-INF/classes` directory to your web application's `WEB-INF/classes/` directory. If you don't have this directory, create it. For example, `<yourWebapp>/WebContent/WEB-INF/classes`
4. Move the `portlets.cfg` file to your domain directory, or if running in a cluster, to a shared directory outside the application.
5. This insures that the directory location will be constant regardless of how the application is deployed.
6. Using a text editor, modify the `default-value` setting in `AutonomyPortletSettings.usr` to point to the `portlets.cfg` file.

Note: You must use an absolute path to point to the `portlets.cfg` file.

7. When finished editing, save your changes.
8. Using a text editor, edit the `portlets.cfg` file and update the following values to point to your IDOL Server port, if changed: `UAPort`, `ClassPort`, and `DREPort`. Edit other settings if needed. The `AutonomyIDOLServer.cfg` is located in the `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<os>/IDOLserver/IDOL` directory.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

9. For more information about configuring this file, see the *Autonomy Portlets for WebLogic Administration Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.
10. Disable JSP fragment validation:
 - a. Right-click your web application project and select **Properties**.
 - b. In the tree, select **Validation - AppXRay**.
 - c. In Validation - AppXRay, deselect **Report Java compilation errors in JSP pages**, and then click **OK**.
 - d. Click **Yes** when asked to rebuild the AppXRay database.
11. When finished editing, save changes. If using Autonomy portlets in the portal application, continue to the next section, [Section 3.1.5, "Enabling Autonomy Portlets in a Portal Application."](#)

Note: To view the portlets, you must add them to your portal application.

3.1.5 Enabling Autonomy Portlets in a Portal Application

When operating in a portal application, Autonomy portlets require the `DownloadServerFileServlet`. You need to register this servlet in your web project by editing the `web.xml` file for your respective web project.

To register this servlet, do the following in Oracle Enterprise Pack for Eclipse:

1. From the Package Explorer, navigate to the respective `web.xml` file for your web project. For example, `<myWebProject>\WebContent\WEB-INF\web.xml`.
2. Right-click your `web.xml` file and select **Open With > XML Editor**.
3. Add the following sections to your `web.xml` file.

- a. Register the servlet:

```
<servlet>
<servlet-name>DownloadServerFileServlet</servlet-name>
<servlet-class>com.bea.apps.groupspace.util.DownloadServerFileServlet</serv
let-class>
</servlet>
```

- b. Register the servlet mapping:

```
<servlet-mapping>
<servlet-name>DownloadServerFileServlet</servlet-name>
<url-pattern>/DownloadServerFileServlet/*</url-pattern>
</servlet-mapping>
```

- c. Register the context parameters:

```
<context-param>

<param-name>com.bea.apps.groupspace.search.enterprise.IDOLServerHost</param
-name>
    <param-value>localhost</param-value>
</context-param>
<context-param>
<param-name>com.bea.apps.groupspace.search.enterprise.IDOL_aciPort</param-n
ame>
    <param-value>9014</param-value>
</context-param>
```

4. To enable basic authentication for administrators (in lieu of a login portlet):

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Site</web-resource-name>
        <url-pattern>/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>Admin</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>myrealm</realm-name>
</login-config>
<security-role>
```

```
<role-name>Admin</role-name>
</security-role>
```

5. Save your `web.xml` file.
6. Re-deploy your application.

3.2 Writing Autonomy-based Applications and Portlets

You can create Autonomy-based applications and portlets using the Autonomy APIs. For Autonomy API documentation, review the Autonomy JavaDoc.

Note: Do not execute queries against any IDOL database which is prefixed with `WLP_CM_REPO` as these indexes contain information on the WLP content repositories in use for your portal. If you want to execute queries against WLP's content management repositories, you need to use the Oracle WebLogic Portal API, see the *Oracle Fusion Middleware Java API Reference for Oracle WebLogic Portal*.

3.3 Creating Autonomy Search Portlets for WLP Content Repositories

You can also create portlets that can be used to search Oracle WebLogic Portal's content management system. To do this, you use the Oracle WebLogic Portal Content Search API. You must also install and configure the BEACMRepoFetch in enable this search capability.

Although you use Autonomy's APIs to create most Autonomy search portlets, you must use Oracle WebLogic Portal Content Search API to create portlets that will search the Oracle WebLogic Portal content management system. See the *Oracle Fusion Middleware Java API Reference for Oracle WebLogic Portal* for more details.

Table 3–3 *Helpful Content Search APIs*

Package/Class	What it does:
<code>com.bea.content.expression.search</code>	This package enables you to build search queries.
<code>FullTextSearchFactory</code>	This class enables you to incorporate full-text search on content within the Virtual Content Repository.

3.4 Configuring Full-Text Wildcard Searches

When performing full-text search wildcard queries for a particular piece of text, finding that text is dependent on the number of occurrences of that text. The default setting is 5. To change this setting, the following needs to be added to the [Server] section of the IDOL Server configuration file (`AutonomyIDOLServer.cfg`):

```
SpellCheckCorrectMinDocOccs=1
```

For example, this query with the default setting:

```
ITextQuery tQuery = FullTextQueryFactory.getTextQuery();
ITextQueryParameter textCriteria = tQuery.buildEquals(new String[]
{ '*Propagation*' });
IFullTextSearch fullTextSearch = FullTextSearchFactory.buildFullTextSearch(null,
textCriteria, false);
Search search = new Search(repositoryPath, -1, null, fullTextSearch, true);
ISearchManager searchManager = ContentManagerFactory.getSearchManager();
IPagedList<Node> nodes = searchManager.search(context, search);
```


receives hits when the word "Propagation" appears in 5 documents.

If the setting:

`SpellCheckCorrectMinDocOccs=2`

is added, the search string only need to appear in 2 documents to received hits.

This setting is also used for spell checking. The Autonomy documentation specifies that "The minimum number of documents that a term has to appear in before IDOL server can use it as a spell check suggestion." For more information, see the Autonomy documentation.

Integrating Your Portal with Autonomy Search

This chapter discusses how to integrate your portal enterprise application to use Autonomy search. To do so, you must configure your application to use the sample Autonomy integration code provided with the Oracle WebLogic Portal installation.

Note: For more information on configuring full text search, see "Adding Full Text Search Capabilities to Oracle WebLogic Portal" in the *Oracle Fusion Middleware Content Management Guide for Oracle WebLogic Portal*

You can also configure Oracle WebLogic Portal to perform full-text searches against content in an Oracle WebCenter Content-based repository. WebCenter Content repositories include built-in support for full text search in Oracle WebLogic Portal. For more information on Oracle Universal Content Management, see *Oracle WebLogic Portal UCM VCR Adapter Guide for Oracle WebLogic Portal*.

If you would like to use Autonomy Enterprise Search, you must first purchase the required license and obtain the binaries from Autonomy Corporation at <http://www.autonomy.com>.

Note: Before integrating the Autonomy sample implementation, you must configure your portal to use Autonomy search. For details, see [Chapter 3, "Configuring Your Portal to Use Autonomy Search."](#)

To configure your application to use the sample Autonomy integration code, you must perform the following procedures in the order listed:

1. [Section 4.1, "Adding the Autonomy Sample Implementation to the Portal Enterprise Application"](#)
2. [Section 4.2, "Modifying the Oracle WebLogic Server Start Script to Start the Autonomy Search Engine"](#)

4.1 Adding the Autonomy Sample Implementation to the Portal Enterprise Application

The Autonomy sample implementation is code that enables Oracle WebLogic Portal to use the Autonomy search engine for full-text search requests against an Oracle WebLogic Portal content repository.

To add the Autonomy sample implementation to the portal enterprise application:

1. Stop Oracle WebLogic Server, if it is currently running.
2. Copy the Autonomy sample implementation file—`autonomy-impl.jar`—and paste the it into your portal enterprise application's `APP-INF/lib` directory.

The `autonomy-impl.jar` file is located in the `<WLPORTAL_HOME>/samples/lib` directory on the machine on which Oracle WebLogic Portal is installed.

Note: If you want to change the default behavior of the sample Autonomy implementation, you can modify the source code, compile it, then add the modified, compiled code—instead of the `autonomy-impl.jar` file—to the portal enterprise application.

The source code is located in the `autonomy-impl-src.zip` file, which is located in the `<WLPORTAL_HOME>/samples/lib/` directory on the machine on which Oracle WebLogic Portal is installed.

3. Navigate to the `content-config.xml` file, located in the portal enterprise application's `META-INF/` directory.
4. Edit the `content-config.xml` file.

Define one search engine manager repository property for each repository that will use Autonomy for full-text search, using the following example code (note that the name of the property must be `cm_searchEngineManager`). The `AutonomyClient46Manager` class is part of the Autonomy integration sample code.

```
<repository-property>
<description>Specify the search engine manager class.</description>
<name>cm_searchEngineManager</name>
<value>com.bea.content.repo.server.logic.search.AutonomyClient46Manager</value>
</repository-property>
```

5. Navigate to the `p13n-config.xml` file, located in the portal enterprise application's `META-INF/` directory.
6. Edit the `p13n-config.xml` file to define an event service listener for the Autonomy integration code, using the following example code:

```
<event-service>
<listener>com.bea.content.repo.server.logic.search.ContentExporterListener</lis
tener>
</event-service>
```

4.2 Modifying the Oracle WebLogic Server Start Script to Start the Autonomy Search Engine

After adding the Autonomy sample implementation to the portal enterprise application, you have the option to modify the Oracle WebLogic Server start script so that it also starts the Autonomy search engine. To do so:

1. Navigate to the Oracle WebLogic Server start script in the portal enterprise application domain's `bin` directory. The Oracle WebLogic Server start script has the following file name:

- LINUX/UNIX: `startWebLogic.sh`
- Windows: `startWebLogic.cmd`

An example path to the Oracle WebLogic Server start script is:

```
<DOMAIN_HOME>/bin/start/startWebLogic.sh
```

2. Edit the Oracle WebLogic Server start script by adding the command to stop the Autonomy search engine, using the following guidelines:

- The Autonomy search engine has the following file name:
 - LINUX/UNIX: `autonomy.sh`
 - Windows: `autonomy.cmd`
- Following is an example of the command used to stop the Autonomy search engine:
 - LINUX/UNIX:


```
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.sh
stop
```
 - Windows:


```
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.cmd
stop
```

Note: The path in the example is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the example. For this reason, you should change the path in your environment accordingly.

- Following is an example fragment of the Oracle WebLogic Server start script, including the command to stop the Autonomy search engine; this command appears in bold. Note that the code in your Oracle WebLogic Server start script might differ from the code included in the example. For this reason, ensure that you add the command to the appropriate line of code in your Oracle WebLogic Server start script.

```
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.sh stop
ALREADY_STOPPED="true"
# Restore IP configuration the node manager starts IP Migration
if [ "${SERVER_IP}" != "" ] ; then
${WL_HOME}/common/bin/wlsifconfig.sh -removeif "${IFNAME}" "${SERVER_IP}"
fi
```

Note: The path in the example is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the example. For this reason, you should change the path in your environment accordingly.

3. Edit the Oracle WebLogic Server start script by adding the command to start the Autonomy search engine, using the following guidelines:

- Following is an example of the command used to start the Autonomy search engine:

- LINUX/UNIX:

```
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.sh
start
```

- Windows:

```
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.cmd
start
```

Note: The path in the example is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the example. For this reason, you should change the path in your environment accordingly.

- Following is an example fragment of the Oracle WebLogic Server start script, including the command to start the Autonomy search engine; this command appears in bold. Note that the code in your Oracle WebLogic Server start script might differ from the code included in the example. For this reason, ensure that you add the command to the appropriate line of code in your Oracle WebLogic Server start script.

```
# Start Autonomy
${WLPORTAL_HOME}/content-mgmt/thirdparty/autonomy-wlp10/autonomy.sh start

JAVA_OPTIONS="${SAVE_JAVA_OPTIONS}"

SAVE_JAVA_OPTIONS=""

CLASSPATH="${SAVE_CLASSPATH}"

SAVE_CLASSPATH=""
```

Note: The path in the example is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the example. For this reason, you should change the path in your environment accordingly.

4. Start Oracle WebLogic Server using the updated Oracle WebLogic Server start script.

The Oracle WebLogic Server start script starts the Autonomy search engine in addition to Oracle WebLogic Server.

Architectural Considerations for Using Autonomy Search

During the architecture phase, you determine what enterprise content you want to make available for your portal and who within your portal environment will be able to search this information.

This chapter includes the following sections:

- [Section 5.1, "Understanding How Autonomy Search is Implemented"](#)
- [Section 5.2, "Deciding What Information to Index"](#)
- [Section 5.3, "Architectural Recommendations"](#)

5.1 Understanding How Autonomy Search is Implemented

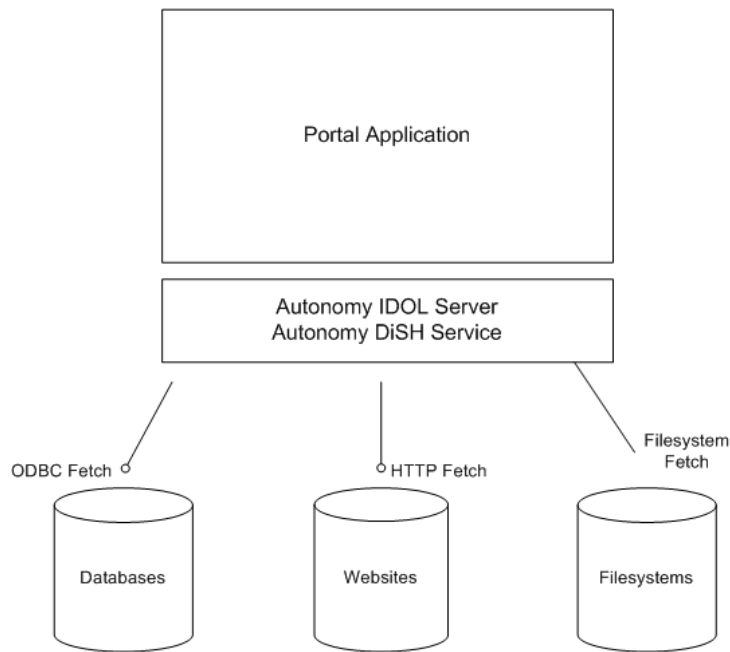
Oracle WebLogic Portal uses Autonomy search components to implement search functions such as allowing portal users to search external web sites, integrated databases, and available file systems. To do this, you can incorporate search portlets that ship with Oracle WebLogic Portal or write your own. However you decide to implement search, the same tools will be used. You will need to install and configure these tools before integrating search within your portal.

Note: If you upgraded to Oracle WebLogic Portal 10.3.4, search portlets are included in your installation. If you are performing a clean install of Oracle WebLogic Portal and want to use Autonomy's search portlets, you must obtain them from Autonomy Corporation.

The Autonomy IDOL server is used to manage the indexes created by the Autonomy fetches you use. You can use this indexed data in your portal application by using the portlets that come with Oracle WebLogic Portal or you can write your own portlets using the Autonomy API. For more information about using and developing portlets, see [Chapter 3, "Configuring Your Portal to Use Autonomy Search."](#)

[Figure 5–1](#) shows a diagram of how Autonomy search tools integrate with Oracle WebLogic Portal.

Figure 5–1 Diagram of an Oracle WebLogic Portal Integration with Autonomy Search Tools



5.2 Deciding What Information to Index

You can allow portal users to search a variety of information sources from your portal. Before developing and deploying your portal, you should decide what information sources you want to index. Consult the respective Autonomy documentation for more details. For example, if you want to include web sites, see the *Autonomy HTTP Fetch Administrator's Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

5.2.1 Maximum Amount of Indexed Content

If you upgraded to Oracle WebLogic Portal 10.3.4, the Autonomy license that is delivered with Oracle WebLogic Portal allows you to index 500,000 pieces of content. A single IDOL server can hold an approximate maximum of 8 million document sections (depending on the functionality and performance required). However, the license included with previous versions of Oracle WebLogic Portal allows only 500,000. The current version of Oracle WebLogic Portal does not include an Autonomy license.

If you performed a clean installation of Oracle WebLogic Portal and purchased Autonomy from Autonomy Corporation, the maximum amount of indexed content available is determined by your license agreement with Autonomy Corporation.

5.3 Architectural Recommendations

Oracle recommends using a separate machine for your Autonomy IDOL Server to ensure the most processing power to service indexing and query requests from your portal clients. You can also install each Autonomy engine (such as HTTP Fetch, File System Fetch and IDOL Server) on a separate server if you find you need additional resources.

Autonomy recommends a dual-processor server for hosting the IDOL Server and the DiSH Handler. For complete system requirements, see the Autonomy documentation.

Configuring Multi-language Searching and Indexing

Oracle WebLogic Portal provides several methods for configuring full-text search and indexing in multiple languages. Each method provides different capabilities. You need to decide on a per-repository basis which method is desirable. If you decided to change methods later, you also need to re-index your repository. Note that each document indexed can be associated with only one language.

The following sections describe each full-text search method and how to configure them:

- [Section 6.1, "One Language per Autonomy Server"](#)
- [Section 6.2, "One Language per Repository"](#)
- [Section 6.3, "Mixing Languages Within a Repository"](#)
- [Section 6.4, "Enterprise Search for Microsoft Word, Excel, and PowerPoint Files in Multibyte Languages"](#)

You need to decide on a per-repository basis which approach is desirable. You should also consult the Autonomy documentation.

6.1 One Language per Autonomy Server

The default configuration for an Autonomy server is one language and one encoding across all repositories using that server. When you use this configuration for multiple languages, you need separate Autonomy servers for each language. In this case you need to configure all indexed content and all full-text queries against that content to use the same `LanguageType` (language and encoding).

For example, you could have three repositories accessing a single Autonomy server. All three repositories must use the same `LanguageType`, such as `FrenchUTF8`, and all documents indexed in each repository would need to be in French. Additionally, all queries on all repositories would need to be in French language with UTF8 encoding. If you needed two languages, you would have to set up two Autonomy servers, two repositories, and manually configure the default language type in each server.

To set a default language type for a server, you edit the `DefaultLanguageType` in the `[LanguageTypes]` section in the server's configuration file (`AutonomyIDOLServer.cfg`). For more information about defining a global default language type, see the *IDOL Server Administration Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

6.2 One Language per Repository

To mix multiple repositories, possibly with different languages, in the same Autonomy server, you need to specify the language and encoding for each repository. This means that all nodes in a repository and all queries must use the same language type and encoding. Both the language type and encoding are defined by the `LanguageType`. Some examples of language types are `frenchUTF8` (French language, UTF8 encoding), `frenchASCII`, and `russianCYRILLIC`. When you use a language type, such as `frenchUTF8`, all documents in the French-UTF8 repository must be in French and all queries in that repository must be in the French language with UTF8 encoding.

The supported language types are listed in `[LanguageTypes]` section in the server's configuration file (`AutonomyIDOLServer.cfg`), which is located in the `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<os>/IDOLserver/IDOL` directory.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

To use this approach, you need to add two properties to the repository configuration. For example, to use the French language with UTF8 encoding add:

- `fullTextSearchIndexLanguageType=frenchUTF8`
- `fullTextSearchQueryLanguageType=frenchUTF8`

Generally you set these properties to the same value. All queries need use the same `fullTextSearchQueryLanguageType` language and encoding.

For instructions on how to add a property to a repository, see "Adding Custom Properties" in the *Oracle Fusion Middleware Content Management Guide for Oracle WebLogic Portal*.

Note: After you disconnect a repository or make any changes to repository properties, Portal Administration Console users must log out and log back in to view the changes.

6.3 Mixing Languages Within a Repository

If you mix data of multiple language types within a repository, you can use Automatic Language Detection. This approach provides the greatest flexibility for both repository content and search options.

Automatic Language Detection identifies the language and encoding of a document when it is indexed and provides the ability to query data by language and/or encoding. For example you could specify that you want to find only French and Italian matches; regardless of encoding; or only Russian matches with UTF8 encoding; or all matches, regardless of language and encoding.

You configure Automatic Language Detection on a per-repository basis. This means you could have three different repositories with different indexing and querying abilities: two repositories might use Automatic Language Detection and have a

mixture of documents of type `frenchUTF8`, `englishASCII`, and `russianCYRILLIC`, while the third repository contains only `italianUTF8` documents.

Caution: When you configure Automatic Language Detection, any repository using the default configuration (one language and one encoding across all repositories using that server), is automatically configured to use Automatic Language Detection. If you do not want this behavior, specify the language type for each language and its encoding for those repositories, as described in [Section 6.2, "One Language per Repository."](#)

6.3.1 Configuring Automatic Language Detection

When Automatic Language Detection is set, the server automatically identifies the language and encoding of a document when it is indexed. For more information about Automatic Language Detection, see the *IDOL Server Administration Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

Note: Enabling this feature may have an impact on the ability to search for existing content in Content Management repositories other than content defined as the `DefaultLanguageType`. This is because language reclassification can occur when this feature is enabled.

To configure Automatic Language Detection on a repository:

1. Set the `AutoDetectLanguagesAtIndex` to `true` in the `[Server]` section of the `AutonomyIDOLServer.cfg` file, which is located in the `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<os>/IDOLserver/IDOL` directory.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

2. Do not set a `fullTextSearchIndexLanguageType` property on the repository; remove if already set.
3. Optionally, specify the default query language type by adding a property to the repository. For example:
4. `fullTextSearchQueryLanguageType=frenchUTF8`
5. Optionally, specify that results are returned across all languages, not just the language of the `fullTextSearchQueryLanguageType` by adding the following property to the repository:
6. `fullTextSearchQueryAnyLanguage=true`
7. Re-index your repository content. For information on how to do this, see [Section 8.2, "Re-Indexing WLP Repository Content."](#)

Note: During indexing, if the language type cannot be determined automatically, the `DefaultLanguageType` is used. This is a global server setting, not a repository setting.

6.3.2 Creating Queries

Queries are very flexible; they can be in any language and encoding. For example, you can construct a query that return results for Japanese documents using UTF-8, Shift_JIS, and EUC-JP encodings.

Use the following examples to specify the search results from your repositories. For additional information about these examples, see the *Oracle Fusion Middleware Java API Reference for Oracle WebLogic Portal*.

6.3.3 Query Text in Same Language and Any Encoding

If the query text is in the language and encoding defined by the `fullTextSearchQueryLanguageType` and you want results in the language of `fullTextSearchQueryLanguageType` regardless of the encoding, you do not need to create additional code.

6.3.4 Query Text in Same Language with Specific Encoding

If the query text is in the language and encoding defined by the `fullTextSearchQueryLanguageType` and you want the results in the same language as the `fullTextSearchQueryLanguageType` with a specific encoding:

```
params = new AutonomyLanguageParameterSet();
params.setLanguageType("englishASCII");
params.setMatchEncoding("UTF8");
context.setParameter(FullTextSearchLanguageParameterSet.
    QUERY_LANGUAGE_PARAMETER_SET_KEY, params);
```

6.3.5 Query Text in Another Language and Encoding

If the query text is in a language and encoding different from `fullTextSearchQueryLanguageType`, you can override the repository `fullTextSearchQueryLanguageType` in the `ContentContext` class. This returns results in the specified `LanguageType` language, regardless of encoding:

```
params = new AutonomyLanguageParameterSet();
params.setLanguageType("englishASCII");
context.setParameter(FullTextSearchLanguageParameterSet.
    QUERY_LANGUAGE_PARAMETER_SET_KEY, params);
```

6.3.6 Query Across All Languages

If the query text is in one language and encoding and you want to query across all languages:

```
params = new AutonomyLanguageParameterSet();
params.setLanguageType("englishASCII");
params.setAnyLanguage(true);
context.setParameter(FullTextSearchLanguageParameterSet.
    QUERY_LANGUAGE_PARAMETER_SET_KEY, params);
```


6.3.7 Query Multiple Specific Languages

If the query text is in one language and encoding and you want to query multiple specific languages:

```
params = new AutonomyLanguageParameterSet();
params.setLanguageType("englishASCII");
params.setAnyLanguage(true);
params.setMatchLanguageType("frenchASCII+germanUTF8");
context.setParameter(FullTextSearchLanguageParameterSet.
    QUERY_LANGUAGE_PARAMETER_SET_KEY, params);
```

6.4 Enterprise Search for Microsoft Word, Excel, and PowerPoint Files in Multibyte Languages

You can configure search and indexing for Microsoft Word (.doc), Excel (.xls), and PowerPoint (.ppt) files in Content Management communities. In these cases, you need to use the default configuration for an Autonomy server, that is, one language and one encoding across all repositories using that server. For more information, see [Section 6.1, "One Language per Autonomy Server."](#)

In addition to using the Autonomy server configuration, you need to set the encodings for indexing and searching on the file names as described in this section. Without these encoding settings, search cannot find the file names based on multibyte encodings. These encoding are set in the following files:

- omnislave.cfg
- AutonomyIDOLServer.cfg

Note: The supported language types and encodings are listed in [LanguageTypes] section in the AutonomyIDOLServer.cfg file, which is located in the <WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<os>/IDOLserver/IDOL directory.

The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

6.4.1 Settings in omnislave.cfg

You must specify the system's default encoding in the omnislave.cfg file. This file is located in the

<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<OS>/filters directory.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path above might differ from the example. For this reason, you should change the path in your environment accordingly.

To specify the encoding:

1. In the `omnislave.cfg` file, remove any `FileNameFromCharSet=<encoding>` settings from any sections in which they appear.
2. In the `[Configuration]` section, add the system's default encoding. For example:

```
FileNameFromCharSet=SHIFTJIS
```

6.4.2 Settings in `AutonomyIDOLServer.cfg`

You must specify the `DefaultLanguageType` and `DefaultEncoding` settings in the `AutonomyIDOLServer.cfg` file. The `DefaultEncoding` must be same encoding as specified in `omnislave.cfg`. And the `DefaultLanguageType` must be in the corresponding language type to the encoding specified in `omnislave.cfg` and the language of document. For example for the Japanese language with Shift-JIS encoding, you would specify:

```
[LanguageTypes]
DefaultLanguageType=japaneseSHIFT_JIS
DefaultEncoding=SHIFTJIS
```

Staging Autonomy Search Functionality

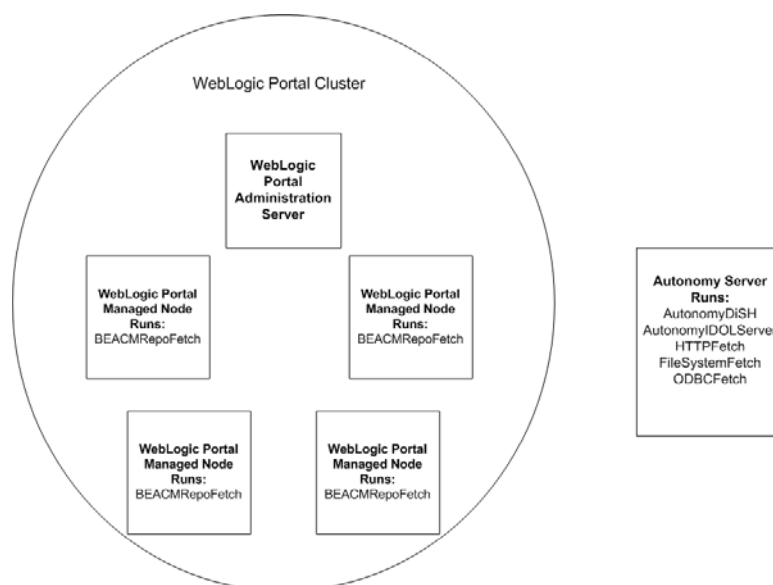
When you move to a production environment, you must configure Autonomy to match the portal environment you use.

This involves editing the configuration files for the search components you are using, deploying the Autonomy Service Dashboard, and configuring Autonomy fetches to search for information according to parameters you set.

The tasks covered in this chapter assume a typical Autonomy configuration running in an Oracle WebLogic Portal cluster. Consult the Autonomy documentation if you want to create a more complex configuration, such as running HTTPFetch on a separate server. Contact Autonomy Corporation to receive the Autonomy documentation.

Figure 7-1 provides an example of a typical production environment.

Figure 7-1 Example of Oracle WebLogic Portal Cluster Using Autonomy



This chapter discusses the following topics:

- [Section 7.1, "Installing Autonomy on Your Target Server"](#)
- [Section 7.2, "Configuring Autonomy on Your Target Server"](#)
- [Section 7.3, "Setting Up Content Management Search"](#)
- [Section 7.4, "Staging File System Fetch within a WebLogic Cluster"](#)

- [Section 7.5, "Starting the Autonomy Services"](#)
- [Section 7.6, "Stopping Autonomy Services in Windows 2000"](#)
- [Section 7.7, "Installing Autonomy Service Dashboard"](#)

7.1 Installing Autonomy on Your Target Server

This section discusses how to install Autonomy on your target server.

7.1.1 Supported Operating Systems

Oracle WebLogic Portal makes available versions of Autonomy that are compatible with operating systems that Oracle WebLogic Portal also supports. However, you may run Autonomy on separate server that uses an operating system that Oracle WebLogic Portal does not support. For more information about the supported configurations for Autonomy, see the Autonomy documentation.

Note: To obtain Autonomy files for a non-Oracle WebLogic Portal supported operating system, contact your Autonomy representative. For more information about configuring Autonomy on a non-WebLogic supported operating system, see [Section 7.3.4, "Configuring the WLP Content Management Fetch When Using a Non-WebLogic Portal Supported Operating System."](#)

7.1.2 Installing Autonomy

This section discusses how to install Autonomy.

Note: This procedure is only valid if you upgraded to Oracle WebLogic Portal 10.3.4. If you performed a clean installation of Oracle WebLogic Portal and obtained Autonomy from Autonomy Corporation, ignore this procedure and perform the installation instructions provided by Autonomy Corporation.

To install Autonomy:

1. Create a directory on your target server for the Autonomy components.
2. From your Oracle WebLogic Portal installation, navigate to the Autonomy distribution for your target operating system. For example,
`<WLPORAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<operatingsystem>`

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

3. Copy the entire directory from your installation directory to the target directory on your target server.

7.2 Configuring Autonomy on Your Target Server

You need to modify your Autonomy configuration to match your production environment and the parameters of your cluster. This includes modifying the respective configurations of the search tools you are using to account for security concerns and their network location.

You also need to configure the types of information you want to include in your searches.

Note: For out-of-the-box implementation, you can set `CONTENT_SEARCH_OPTION=full` in the `startWebLogic` script and add the necessary properties to the repository.

[Table 7–1](#) lists the location of the configuration files you need to modify.

Table 7–1 *Autonomy Components and Their Respective Configuration Files*

Autonomy Component	Configuration File You Need to Modify
DiSH Server	<code>//IDOLserver/DiSH/AutonomyDiSH.cfg</code>
IDOL Server	<code>//IDOLserver/IDOL/AutonomyIDOLServer.cfg</code>
Agent Stores for the IDOL Server	<code>//IDOLserver/IDOL/agentstore/agentstore.cfg</code>
HTTP Fetch	<code>//HTTPFetch/HTTPFetch.cfg</code>
File System Fetch	<code>//FileSystemFetch/FileSystemFetch.cfg</code>

This section includes the following topics:

- [Section 7.2.1, "Configuring the Autonomy IDOL Server"](#)
- [Section 7.2.2, "Configuring the Autonomy DiSH"](#)
- [Section 7.2.3, "Configuring Agentstore"](#)
- [Section 7.2.4, "Configuring HTTP Fetch"](#)
- [Section 7.2.5, "Configuring File System Fetch"](#)

7.2.1 Configuring the Autonomy IDOL Server

The Intelligent Data Operating Layer (IDOL) server is responsible for indexing content as well as processing queries. For more information about the IDOL server, see the Autonomy IDOL Server documentation.

To configure the IDOL server for your production environment modify the `AutonomyIDOLServer.cfg` file. The file is located `//autonomy/IDOLserver/IDOL/AutonomyIDOLServer.cfg`.

To configure the Autonomy IDOL server:

1. Open the `AutonomyIDOLServer.cfg` file in a text editor.
2. In the `[License]` section, edit the `LicenseServerACIPort` to match the port on which DiSH is running, if you changed this port.
3. In the `[Service]` section, enter the port number by which service commands can be sent to DiSH. By default this is 20003. Note that this port must not be used by any other service.

4. In the `[Server]` section,
 - Edit the client list settings (`IndexClients`, `AdminClients`) for security as required.
 - Edit the `IndexPort` and `Port` (ACI) settings as needed.
5. In the `[Paths]` section, edit the `Modules` and `TemplateDirectory` to point to the location of these directories on the target system. These must be absolute paths.
6. Locate and edit all other directory or file path settings and adjust to point to the new location (for example, the `[NT_V4] Library`). These must be absolute paths.
7. In the `[Database]` section, create and remove Autonomy databases as required for your needs. Consult the Autonomy documentation for managing databases.
8. When finished, save your changes.

7.2.2 Configuring the Autonomy DiSH

The Distributed Services Handler (DiSH) is used to manage Autonomy components. You can access DiSH functions through the Autonomy Service Dashboard or use Autonomy's ACI interface. For more information about the Autonomy DiSH, see the Autonomy DiSH documentation.

To configure DiSH, you can use a text editor to modify the `autonomyDiSH.cfg` file. The `autonomyDiSH.cfg` file is located in the `//autonomy/IDOLserver/IDOL/AutonomyIDOLDiSH.cfg` directory.

To configure Autonomy DiSH:

1. Open the `autonomyDiSH.cfg` file in a text editor.
2. In the `[Service]` section, edit the `ServicePort` setting if needed to avoid port conflicts.
3. In the `[Server]` section, edit the following:
 - Modify the `AdminClients` as required for establishing security as needed.
 - Modify the `Port` setting if needed to avoid port conflicts.
4. In the `[Email]` section, make modifications as defined by your company's SMTP setup.
5. In the `[ChildServices]` section, remove the setting for the `BEACMRepoFetch` service.
6. Remove the `[BEACMRepoFetch]` section.
7. In the `[IDOLServer]`, `[HTTPFetch]` and `[FileSystemFetch]` sections, modify each path to ensure the executable files use the location on the target server. These paths must be absolute.
8. If you changed the `Service Port` or ACI port (or plan on doing so in the `agentstore.cfg` file), you need to adjust these settings to match.
9. When finished, save your changes.

7.2.3 Configuring Agentstore

Agents provide the facilities to find and monitor information from a configurable list of internet and intranet sites, news feeds, chat streams and internal repositories that you want to enable your portal users to search.

For more information about using agents, see the *Autonomy IDOL Server Guide*, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

To configure the Agentstore for your cluster, edit the `agentstore.cfg` file. The file is located `//IDOLserver/IDOL/agentstore/agentstore.cfg`.

To configure agentstore:

1. Open the `agentstore.cfg` file in a text editor.
2. Modify `[License]`, `[Service]` and `[Server]` settings as required for port conflicts and security.
3. Locate and replace all file and directory settings and adjust to point to the new location. The paths must be absolute.
 - In the `[Paths]` section, change the `TemplateDirectory` to point to the new location.
 - In the `[Logging]` section, change the `LogDirectory` and the `LogArchiveDirectory` to point to the new location.
 - In the `[LanguageTypes]` section, change the `LanguageDirectory` to point to the new location.
4. When finished, save your changes.

7.2.4 Configuring HTTP Fetch

HTTP Fetch is responsible for crawling specified web sites and passes the content to the IDOLServer for indexing. You need configure this fetch and create HTTP fetch jobs that you need.

You do this by editing the `HTTPFetch.cfg` file. It is located `//autonomy/HTTPFetch/HTTPFetch.cfg`

To configure the HTTP Fetch:

1. Open the `HTTPFetch.cfg` in a text editor.
2. The `[Service]` section determines which machines are permitted to use and control the HTTPFetch service via the service port. Modify the port and client security control as required.

Note: If you modify the port settings in this file, you need to update the HTTPFetch port settings in the `AutonomyDiSH.cfg` file.

3. The `[Default]` section contains the default settings that apply to all the jobs that you define in `[Spider]` section. If you changed the `IndexPort` in the `AutonomyIDOLserver.cfg` file, you need to modify the `IndexPort` setting to match.
4. When finished, save your changes.

5. Create HTTP Fetch jobs as required (to spider and index the web sites you want to search). For information about creating fetch jobs, see the [Autonomy HTTPFetch documentation](#).

7.2.5 Configuring File System Fetch

File System Fetch polls specified areas of a file system and, when content changes are found, imports the content and passes the content to the IDOLServer for indexing.

To control how files are imported from an internal location (for example, from a computer on your network), you need to configure File System Fetch and then create the fetch jobs you need.

The File System Fetch configuration file is located:

```
//autonomy/FileSystemFetch/FileSystemFetch.cfg
```

Use a text editor to edit the `FileSystemFetch.cfg` file to match your production environment.

To configure File System Fetch:

1. Modify the `[Server]` and `[Service]` sections to change ports, if needed, and to control security. If you modify the port information in this file, you need to also update the settings related to File System Fetch in the `AutonomyDiSH.cfg` file.
2. In the `[Default]` section, modify the `IndexPort` to match the `IndexPort` set in `AutonomyIDOLserver.cfg`, if necessary.
3. When finished, save your changes.
4. Create File System Fetch jobs as required (to spider and index certain file system locations) for your needs. For information about creating fetch jobs, see the [Autonomy File System Fetch documentation](#).

If deploying Oracle WebLogic Portal in a cluster environment, you need to ensure that each machine in your cluster can access the content indexed by File System Fetch, see [Section 7.4, "Staging File System Fetch within a WebLogic Cluster."](#)

7.3 Setting Up Content Management Search

To set up full-text search for your WLP repositories, you must configure the WLP Content Management fetch and then enable your WLP repositories for full-text search.

This section includes the following topics:

- [Section 7.3.1, "Configuring the Content Management Fetch"](#)
- [Section 7.3.2, "Configuring WLP Repositories for Full-Text Search"](#)
- [Section 7.3.3, "Troubleshooting Full-Text Search for WLP Content Management Repository"](#)
- [Section 7.3.4, "Configuring the WLP Content Management Fetch When Using a Non-WebLogic Portal Supported Operating System"](#)
- [Section 7.7.1, "Prepare the Dashboard for Installation"](#)
- [Section 7.7.2, "Deploy the Autonomy Service Dashboard"](#)

7.3.1 Configuring the Content Management Fetch

The content management fetch enables full-text search for WLP repositories. For each managed server in your Oracle WebLogic Portal cluster, you need to configure the content management fetch.

To configure the content management fetch:

1. Set an environment variable called `CONTENT_SEARCH_OPTION` and assign it a value of `minimal`.
2. Edit the `BEACMRepoFetch.cfg` file. This file configures the settings for the full-text search of repositories.

If you have upgraded to Oracle WebLogic Portal 10.3.4, the `BEACMRepoFetch.cfg` file is located in `//operating_system_directory/internal/BEACMRepoFetch/BEACMRepoFetch.cfg`.

If you have performed a clean installation of Oracle WebLogic Portal 10.3.4, you must create your own `BEACMRepoFetch.cfg` file—using the code in [Example 7–1, "Example BEACMRepoFetch.cfg File"](#)—then add the file to the `//operating_system_directory/internal/BEACMRepoFetch` directory.

Perform the following edits to the `BEACMRepoFetch.cfg` file:

- Modify `[Server]` and `[Default]` settings to change port numbers and client security as required.
 - Modify `[Default]` `DreHost` settings to point to the hostname or IP address of the server which is running the IDOLServer.
 - Modify `[Default]` `IndexPort` to match the `IndexPort` setting in the `AutonomyIDOLServer.cfg` file on the remote server.
3. When finished, save your changes.

Note: Do not modify any other settings within this file.

Example 7–1 Example BEACMRepoFetch.cfg File

If you have performed a clean installation of Oracle WebLogic Portal, you should create a `BEACMRepoFetch.cfg` file—using the code in this example—then add the file to the `//operating_system_directory/internal/BEACMRepoFetch` directory.

```
[Server]
Port=9110
QueryClients=*
AdminClients=*
Threads=5

[Service]
ServicePort=10084
ServiceControlClients=*
ServiceStatusClients=*

[Default]
AciPort=9014
PollingAction=2
PollingMaxNumber=1000
DreHost=127.0.0.1
```

```
IndexPort=9001
PollingMethod=2
PollingPostAction=1
PollingPeriod=5 seconds
AllowOriginalFileDeletion=true
RemoveLogFileOnStart=on
ImportIDXFilesAction=0
ImportTempDir=./importTemp
ImportDefaultSlaveDirectory=../../filters
ImportCharsetConvTablesDirectory=../../IDOLserver/IDOL/langfiles
ImportReadChecking=true
StableCheckMinWaitTime=2
IndexMode=cmUniqueName

[Configuration]
Number=2
0=BEACMRepoImport
1=BEACMRepoIDXImport

[BEACMRepoImport]
DirectoryPathCSVs=../../internal/BEACMRepoTemp/binary
DirectoryRecurse=on
ImportStoreContent=on
ImportSummary=on
ImportBreaking=ON
ImportBreakingMinParagraphWords=300
ImportBreakingMaxParagraphWords=500
ImportBreakingMinDocWords=500
ImportIntelligentTitleSummary=0
ImportExtractDateFrom=8
ImportExtractDateToField=DREDATE
ImportExtractDateToFormat=EPOCHSECONDS

[BEACMRepoIDXImport]
DirectoryPathCSVs=../../internal/BEACMRepoTemp/nonbinary
DirectoryFileMatch=*.txt,*.idx
DirectoryRecurse=on
ImportMinLength=0
ImportMinLengthWords=0
ImportStoreContent=off
ImportBifIncludeQuotes=true
```

7.3.2 Configuring WLP Repositories for Full-Text Search

After you have configured the Content Management Fetch, you need to enable your WLP repositories to take advantage of full-text search. This ensures your WLP repositories can locate the Autonomy IDOL server.

This section includes the following topics:

- [Section 7.3.2.1, "Configuring Oracle WebLogic Portal Repository Autonomy Sample Integration Properties"](#)
- [Section 7.3.2.2, "Editing Full-Text Search Properties"](#)

7.3.2.1 Configuring Oracle WebLogic Portal Repository Autonomy Sample Integration Properties

You need to define Autonomy sample integration properties for your WLP repositories within the Virtual Content Repository using the Portal Administration

Console. These properties ensure that your WLP repositories can locate the Autonomy services.

[Table 7–2](#) lists the Autonomy sample integration properties you need to configure.

Table 7–2 Autonomy Sample Integration Properties for WLP Repositories

Property	Definition
public.search.staging.area	<p>You need to set this property ONLY if you are using using a shared drive to index content. For more information, see Section 7.3.4, "Configuring the WLP Content Management Fetch When Using a Non-WebLogic Portal Supported Operating System."</p> <p>When setting this property, you must use the system default file delimiter or the data will not be properly indexed, as shown in the following examples:</p> <ul style="list-style-type: none"> Windows: public.search.staging.area=\\...\\cm\\thirdparty\\autonomy-wlp10\\internal\\BEACMRepoTemp UNIX: public.search.staging.area=/.../cm/thirdparty/autonomy-wlp10/internal/BEACMRepoTemp <p>Note: The paths with wlp10 above and below are based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the paths above and below. For this reason, you should change the paths in your environment accordingly.</p> <p>Note: This path is appended to the <WLPORTAL_HOME> directory and therefore needs to be relative to that directory. When you set this path, be sure to start the path with a file separator character and use slashes appropriate for your operating system. Also be sure all directories in the path from wlportal_10.3 exist.</p> <p>The default directory is:</p> <pre>/content-mgmt/thirdparty/autonomy-wlp10/internal/BEACMRepoTemp</pre>
public.search.engine.host	This is the hostname for the machine on which the IDOL server resides.
public.search.index.port	<p>This is the Autonomy index port.</p> <p>This value needs to match the [Server] IndexPort setting in the AutonomyIDOLServer.cfg file. See Section 7.2.1, "Configuring the Autonomy IDOL Server" for information about this file.</p>
public.search.query.port	<p>This is the port setting that is used by the IDOL server.</p> <p>This value needs to match the [Server] Port setting in the AutonomyIDOLServer.cfg file. See Section 7.2.1, "Configuring the Autonomy IDOL Server" for information about this file.</p>
public.search.urlconnection.timeout	When Autonomy database commands are issued using HTTP to the search indexing port and the search engine port, this time-out setting specifies the HTTP connection time-out, in milliseconds. The default time-out is 180000 (180 seconds).

Note: After you make any changes to repository properties, Portal Administration Console users must log out and log back in to view the changes.

To add a property to a repository:

1. From the main menu of the Portal Administration Console, select **Content > Content Management**.
2. In the resource tree, click **Repositories** to view the Manage | Repositories tree.
3. In the Manage | Repositories resource tree, select the WLP Repository to which you want to add a property.
4. In the Properties section on the Summary tab, click **Add Property**.
5. In the Add Property dialog, enter the name and value for your property. Enter each property included in [Table 7-2](#).
6. Click **Save**.

A summary of the new repository information is displayed in the Summary tab.

7.3.2.2 Editing Full-Text Search Properties

You need to ensure that all full-text functions are enabled for each WLP repository you want to enable to use full-text search.

[Table 7-3](#) lists the advanced full-text search repository properties and how they are used.

Table 7-3 Required Settings for Full-Text Search

Advanced Property	What it does:
Search Enabled	Enables users to search the repository.
Search Indexing Enabled	For a repository that supports full-text search, allows content to be indexed for portal search. This enables portal developers to include full-text content search or metadata search in any portlets that they develop.
Full-Text Search Enabled	For a repository that supports full-text search, enables users to search the repository using the full-text of the content within the repository.

To edit full-text search repository properties:

1. Select **Content > Content Management** from the navigation menu at the top of the console.
2. Select **Manage | Repositories**.
3. In the resource tree, click the repository you want to modify to view its Summary tab.
4. In the Advanced section, click **Advanced** to view the Edit Advanced Properties for Repository dialog.
5. In the Edit Advanced Properties for Repository dialog, ensure that each property in [Table 7-3](#) is enabled.
6. When finished making changes, click **Save**.

Your modifications display in the Advanced section of the Summary page.

Note: After you disconnect a repository or make any changes to repository properties, Portal Administration Console users must log out and log back in to view the changes.

7.3.3 Troubleshooting Full-Text Search for WLP Content Management Repository

Use the following to check full-text search in your Autonomy configuration:

- Verify that the Autonomy processes are running: AutonomyDiSH.exe, content.exe, and so on.
- Verify that the data is indexed in Autonomy. To view all data in Autonomy, use `http://localhost:9014/action=list`.
- Verify that the repository configuration settings are enabled:
 - `cm_fireRepositoryEvents=true`
 - `search-is-enabled=true`
 - `search-indexing-is-enabled=true`
 - `fulltext-search-is-enabled=true`
- Verify that the `ObjectClass` is marked searchable.
- Verify that `ObjectClass` property definitions are marked searchable.

Note: For more information about `ObjectClass`, see the *Oracle Fusion Middleware Java API Reference for Oracle WebLogic Portal*.

- Verify that Autonomy is configured to start automatically. This option is in the `domain/bin/startWebLogic` script.
- Scan the Autonomy log files (files ending in `.log`) for warnings and errors. These files are under
`<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/...`
 directory.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

- Verify that indexed data is being written to the `FileSystemFetch` directory specified via the `public.search.staging.area` repository configuration property. The default tree is under
`<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/internal/BEACMRepoTemp`.

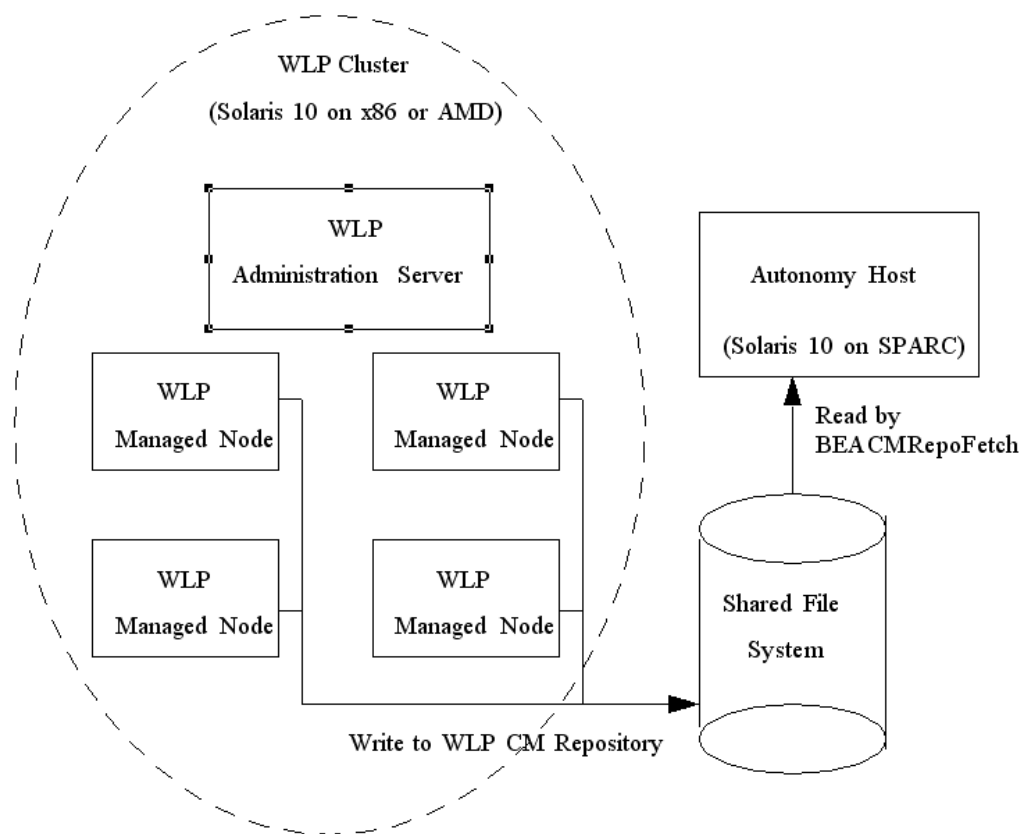
Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

7.3.4 Configuring the WLP Content Management Fetch When Using a Non-WebLogic Portal Supported Operating System

If you run Autonomy on an operating system that is not also supported by Oracle WebLogic Portal, you must configure the content management search differently. You must create a shared file system that can be written to by Oracle WebLogic Portal and also accessed by Autonomy's server.

Figure 7–2 provides an example of a remote Autonomy installation using a shared file system.

Figure 7–2 Example Remote Autonomy Installation on a non-Oracle WebLogic Portal supported operating system



To configure WLP Content Management search:

1. Stop all Autonomy services.
2. Create a shared directory where `shared_drive` is the name of your shared drive.
3. On the Autonomy host, mount `shared_drive`.

4. For each managed server in your cluster, mount `shared_drive`.

Note: Mount `shared_drive` with the same exact mapping on each managed server.

5. On each managed server, set the `CONTENT_SEARCH_OPTION` environment variable to `none`. This prevents Autonomy from starting the content management search.
6. Using the Oracle WebLogic Portal Administration Console, define a repository property called `public.search.staging.area` with a value of `shared_drive`. For more information on setting other Autonomy properties, see [Section 7.4, "Staging File System Fetch within a WebLogic Cluster."](#)
7. Use a text editor to modify the
`<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/<operating_system_directory>/internal/BEACMRepoFetch/BEACMRepoFetch.cfg` file to point to the `shared_drive` you have created. This will ensure that the repository content gets indexed.
 - In the `[BEACMImport]` section, set the `DirectoryPathCSVs` variable to match the directory of your `shared_drive/binary`
 - In the `[BEACMRepoIDXImport]` section, set the `DirectoryPathCSVs` variable to match the location of your `shared_drive/nonbinary`.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

8. Restart Autonomy services. See the `autonomy.sh` or `autonomy.cmd` file in `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10` for a sample start script.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

7.4 Staging File System Fetch within a WebLogic Cluster

When you deploy Oracle WebLogic Portal in a cluster environment, each machine in the cluster must be able to access information and content that is indexed by Autonomy fetches. For example, both WLP repositories and Autonomy's File System Fetch use file systems to store indexed content. You should configure each machine in your cluster to be able to access these file systems.

Note: The Content Management Fetch does not require these steps, see [Section 7.3, "Setting Up Content Management Search."](#)

File System Fetch is used to index content that resides in a file system. When indexing, unless otherwise configured, the `DREREFERENCE` property is set to the complete path of the file. Therefore, with default queries, the link to return the actual content (file) will be the path to the file. Within a server cluster, each node in the cluster must have access to the file system on which the document resides.

1. Create a shared file system that is accessible by both the host machine upon which File System Fetch resides and also accessible by each node in your Oracle WebLogic Portal cluster. The mapping to the path where the files reside must be the same for each node in the cluster and the `FileSystemFetch` host.
2. Place the files to be imported/indexed into the shared drive as required.
3. Configure the `FileSystemFetch` job to import/index the contents of the shared drive using the mapping from the above step. For more information about configuring File System Fetch, see the Autonomy File System Fetch documentation.

Note: When returning query results to the browser and displaying a link to access/download the file, pass the `DREREFERENCE` property (which will contain the fully qualified path/file name) through a servlet which will stream the file to the browser. For more information about indexing and queries, see the *Autonomy IDOL Server Guide* and the Autonomy JavaDoc, published by Autonomy Corporation. Contact WebLogic Portal Customer Support to obtain a copy of this guide.

7.5 Starting the Autonomy Services

You must configure the start script that is used to start Autonomy services on your server. You can either copy these to your target server and modify as required, based on your target directory or you can create similar scripts to meet your needs.

The Autonomy start script depends on two environment variables that should be set on your portal domain server: `WL_` and `CONTENT_SEARCH_OPTION`.

- `WL_` must be set to the weblogic directory in your WebLogic installation
- `CONTENT_SEARCH_OPTION` is used to indicate that level to which Autonomy will run.
 - `full` indicates to run the Autonomy DiSH engine and have it start all configured children
 - `minimal` indicates to only run the `BEACMRepoFetch` engine for use within the WLP repositories (this is the minimal production configuration when using WLP repositories)
 - `none` disables all Autonomy services.

You call this script with either `start` or `stop` as a parameter.

1. Review the `autonomy.cmd/sh` files that reside in the `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10` directory. [Example 7-2](#) shows an example script.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

2. Modify as necessary. Be sure to map the Autonomy shared library directories and ensure that the `AutonomyDiSH.exe` is started.

Example 7-2 Example Autonomy Start Script

```
setlocal

if "%WL_" == "" goto :NO_WL_
if "%1" == "" goto :USAGE
cd %WL_%\cm\thirdparty\autonomy-wlp10\win32
if "%1" == "start" (
    echo Cleaning up license and uid files
    rmdir /s /q HTTPFetch\license >nul 2>&1
    rmdir /s /q IDOLserver\DiSH\license >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\content\license >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\agentstore\license >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\category\license >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\community\license >nul 2>&1
    rmdir /s /q HTTPFetch\uid >nul 2>&1
    rmdir /s /q internal\BEACMRepoFetch\uid >nul 2>&1
    rmdir /s /q IDOLserver\DiSH\uid >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\uid >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\content\uid >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\agentstore\uid >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\category\uid >nul 2>&1
    rmdir /s /q IDOLserver\IDOL\community\uid >nul 2>&1
    rmdir /s /q FileSystemFetch\uid >nul 2>&1
    echo Starting Autonomy with CONTENT_SEARCH_OPTION = %CONTENT_SEARCH_OPTION%
    if "%CONTENT_SEARCH_OPTION%" == "full" (
        if not exist IDOLserver\DiSH\AutonomyDiSH.exe (
            @echo Unable to locate the Autonomy DiSH executable. Cannot start
                the search engine.
        goto :_the_end
        )
        cd IDOLserver\DiSH
        %WL_%\server\bin\beaexecg.exe -hidewindow -command:"AutonomyDiSH.exe"
        @echo Autonomy Distributed Search Handler engine started.
    )
    if "%CONTENT_SEARCH_OPTION%" == "minimal" (
        if not exist internal\BEACMRepoFetch\BEACMRepoFetch.exe (
            @echo Unable to locate the BEACMRepoFetch executable. Cannot start
                the search engine.
        goto :_the_end
        )
        cd internal\BEACMRepoFetch
        %WL_%\server\bin\beaexecg.exe -hidewindow -command:"BEACMRepoFetch.exe"
        @echo Autonomy BEACMRepoFetch engine started.
    )
    goto :_the_end
)
if "%1" == "stop" (
```

```
@REM taskkill depends on the path to WBem. Adding it here
@REM just to ensure that it exists on the system path.
set PATH=%SystemRoot%\System32\Wbem;%PATH%

if "%CONTENT_SEARCH_OPTION%" == "minimal" (
    taskkill /F /T /IM BEACMRepoFetch* >nul
    @echo Autonomy BEACMRepoFetch engine stopped.
)
if "%CONTENT_SEARCH_OPTION%" == "full" (
    taskkill /F /T /IM AutonomyDiSH* >nul
    @echo Autonomy processes stopped.
)
goto :_the_end
)
goto :USAGE
:NO_WL_
@echo The environment variable WL_ is not set. Cannot start Autonomy DiSH.
goto _the_end
:USAGE
@echo Usage: "autonomy.cmd [start|stop]"
pause
goto _the_end

:_the_end
endlocal
```

Note: The path in the example above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

1. Run your script.
2. Verify that the services are running. On Windows, use the TaskManager application and view the Processes tab. If using Unix, use the `ps` command to view a list of services that are currently running. The following services should be running:
 - `content.exe`
 - `category.exe`
 - `community.exe`
 - `agentstore.exe`
 - `AutonomyIDOLserver.exe`,
 - `AutonomyDiSH.exe`
 - `HTTPFetch.exe`
 - `FileSystemFetch.exe`.
3. Inspect the log files for each of the services to verify that there were no errors such as port conflicts, license restrictions, and so on.

7.6 Stopping Autonomy Services in Windows 2000

On a Windows 2000 host, content management full-text search requires the PsKill utility to be installed and available in the PATH. PsKill is required to properly shut down the full-text search processes when the domain is shut down. The PsKill utility can be found at:

<http://www.microsoft.com/technet/sysinternals/ProcessesAndThreads/PsKill.msp>.

After installing it, run the PsKill command once to accept the license. Make sure it exists on your system PATH, and then restart the server. The scripts can now properly shut down the full-text search processes.

7.7 Installing Autonomy Service Dashboard

The Autonomy Service Dashboard is a stand-alone front-end web application that allows administrators to manage all Autonomy modules and child services running locally or remotely.

The Dashboard communicates with one or more Autonomy Distributed Service Handler (DiSH) modules that provide the back-end process for monitoring and controlling all the Autonomy child services, such as fetches.

You deploy the Autonomy Service Dashboard as a portal application within your enterprise application using the Oracle WebLogic Server Console. Before deploying the dashboard, you must modify the configuration to match your production environment.

For complete documentation on how to use the Autonomy Service Dashboard, see the Autonomy DiSH documentation.

This section includes the following topics:

- [Section 7.7.1, "Prepare the Dashboard for Installation"](#)
- [Section 7.7.2, "Deploy the Autonomy Service Dashboard"](#)

7.7.1 Prepare the Dashboard for Installation

Before you deploy the Autonomy Service Dashboard, you need to edit the location configuration to match the new deployment location. The default location that is used is:

```
<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/common/lib/.
```

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

To prepare the Autonomy Service Dashboard for installation:

1. Copy the `automy servicedashboard.cfg` to its new location.
2. Edit the configuration information to match the new location by editing the `web.xml` file for the Autonomy Service Dashboard.

- a. Create a temporary directory to use when completing your edits. For example, `c:/temp/working`
- b. Copy the `autonomyservicedashboard.war` to your working directory.
- c. Using a compression utility such as WinZip or JavaJar, unzip the `autonomyservicedashboard.war`.
- d. Configure the `WEB-INF/web.xml` file by editing the `<context-param>` value to match the new location of the `autonomyservicedashboard.war` file. The default location is `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/common/lib`.
- e. Save the `C:/working/META-INF/web.xml` file.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

3. In the `C:/temp/working` directory, re-zip or re-compress the `autonomyservicedashboard.war` file. Overwrite the existing `autonomyservicedashboard.war` file with the new file of the same name that contains your modified `web.xml` file. Be sure to keep the files in their original directory structure, including the `META-INF` directory.
4. Copy the `autonomyservicedashboard.war` file you just created back to the location where you want to locate the Autonomy Service Dashboard. For example, `<WLPORTAL_HOME>/content-mgmt/thirdparty/autonomy-wlp10/common/lib/`.

Note: The path above is based on an upgraded Oracle WebLogic Portal domain. If you performed a clean installation of Oracle WebLogic Portal and installed Autonomy separately from Oracle WebLogic Portal, the path in your environment might differ from the path above. For this reason, you should change the path in your environment accordingly.

7.7.2 Deploy the Autonomy Service Dashboard

You should deploy the Autonomy Service Dashboard in the same domain as is used by your portal cluster. However it needs to be deployed as a stand-alone application rather than part of your portal application.

Note: You can also deploy the Autonomy Service Dashboard into another web application container, such as Tomcat.

After you deploy the Autonomy Service Dashboard, you can use the default user name of `admin` and the default password of `admin` to log in.

To deploy the Autonomy Service Dashboard:

1. From the Oracle WebLogic Portal domain where you wish to deploy the Autonomy Service Dashboard, run the Oracle WebLogic Server Console.
2. In the Domain Structure section, select **Deployments**. This displays a list of the deployed components.
3. In the Change Center section, click **Lock & Edit**.
4. In the Summary of Deployments section, click **Install**.
5. Navigate to the location of the `autonomyservicedashboard.war` file and select it.
6. Click **Next**.
7. Select **Install this deployment as an application** and click **Next**.
8. Make changes as required
9. Click **Finish**.
10. In the Summary of Deployments section, review any messages or errors displayed.
11. In the Change Center section, click **Activate Changes**.
12. Verify that the `autonomyservicedashboard` deployment is prepared.
13. Mark the check box next to the **autonomyservicedashboard** deployment and click **Start > Servicing all requests**.
14. In the Start Application Assistant, click **Yes**.
15. Navigate to the Autonomy Service Dashboard to verify that it is deployed. The default URL is: `http://localhost:7001/autonomyservicedashboard`.
16. Using the default login (`admin`) and password (`admin`), log in to the Autonomy Service Dashboard.
17. Configure the Autonomy Service Dashboard to point to your DiSH implementation (use the same server and port settings that you used when you edited the `autonomyDiSH.cfg` file in [Section 7.2.2, "Configuring the Autonomy DiSH."](#))
18. For complete documentation on how to use the Autonomy Service Dashboard, see the Autonomy DiSH documentation.
19. Ensure that the IP address of the computer(s) running the Autonomy Service Dashboard are configured in the AdminClient settings of the services. For information on how to configure these settings, see the Autonomy IDOL Server documentation.

Managing Autonomy Search in Production

After you have deployed your portal, you can manage your search services.

This chapter includes the following sections:

- [Section 8.1, "Using the Autonomy Service Dashboard"](#)
- [Section 8.2, "Re-Indexing WLP Repository Content"](#)

8.1 Using the Autonomy Service Dashboard

The Autonomy Service Dashboard allows you to access the Autonomy DiSH server, which monitors the performance of Autonomy's search services.

You can also monitor services using Autonomy's ACI interface. For more information about monitoring search services, see the Autonomy DiSH documentation.

8.2 Re-Indexing WLP Repository Content

You can re-index WLP repository content at any time. For example, you may need to re-index WLP content if your indexes get corrupted (power outage, hardware problems, and so on).

WebLogic Portal provides a script that you can use to re-index WLP repository content. You can either use command line arguments or a `.properties` file to indicate the content you want to index.

Note: Oracle WebLogic Server must be running when re-indexing content.

To re-index content, do the following:

1. From any managed server in your cluster, navigate to the `index_cm_data.cmd/sh` script. It is located in the `<WLPORTAL_HOME>/content-mgmt/bin` directory.

Tip: You can view help for re-indexing content by typing `index_cm_data -help`.

2. Optionally, if using the `cm_indexer.properties` file, modify the properties file to match the parameters of your configuration. [Example 8-1](#) provides an example of the `cm_indexer.properties` file.

- Run the script. [Table 8–1](#) provides a complete listing of the command line arguments and their descriptions. If no arguments are set, the `cm_indexer.properties` is assumed and used. See [Example 8–2](#) for an example of a `cm_indexer.properties` file. If using command line arguments, see [Example 8–3](#) for an example.

Example 8–1 Example of Using the `cm_indexer.properties` File

```
C:\<MW_HOME>\wlportal_10.3\content-mgmt\bin
```

Example 8–2 Sample `cm_indexer.properties` File

```
# Set verbose to true if you want to view any error messages.
verbose=true
#Use username and password that is used to access the portal application.
user=weblogic
password=weblogic
#Use the t3 protocol to refer to the WebLogic Server URL
url=t3://localhost:7001
#Indicate the name of the repository
repository=Shared Content Repository
#Indicate the name of the portal application
application=portalApp
#Optionally, indicate which content types you want to index.
type=
#Indicate the repository path of the content you want to index.
path=/Shared Content Repository
```

Example 8–3 Example of Using Command Line Arguments:

```
C:\myhome\wlportal_10.3\content-mgmt\bin\index_cm_data -verbose -user weblogic
-password weblogic -url t3://localhost:7001 -repository myRepo -application
myPortalApp -path \myRepo
```

Table 8–1 Command Line Arguments for the `cm_index_data` Script

Argument	Description
verbose	Set verbose to true to view error messages.
user	The user name.
password	The user password.
url	The URL of the WebLogic Server. For example, when running the <code>CM_INDEX_DATA</code> script for the local machine, this URL should be: <code>t3://localhost:7001</code> . Note that you should use the T3 protocol, not HTTP.
repository	Name of the repository you want to index.
application	Name of the portal application that uses the repository.
type	Optional. Indicate which content type you want to index.
path	The path of the repository or repository folders you want to index. For example, if you want to index the entire repository use <code>path=/RepositoryName</code> . If you want to index a particular folder within the repository, use <code>path=/RepositoryName/FolderName</code> .