



Oracle Knowledge Intelligent Search Administration Guide

Configuring and Administering Oracle Knowledge Intelligent Search

Oracle Knowledge Version 8.5.1
Document Number OKIS-ADMN851-03
June 2013

Oracle, Inc.

COPYRIGHT INFORMATION

Copyright © 2002, 2013, Oracle and/or its affiliates. All rights reserved.

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. Other names may be trademarks of their respective owners.

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	About This Guide	1
	In This Guide	1
	Screen and Text Representations	2
	References to Web Content	2
	Examples of Product Screens and Text	2
	Operating System Variations in Examples and Procedures	2
Chapter 1	Oracle Knowledge Application Overview	3
	Oracle Knowledge Components	3
	Application Configuration and Management Tools	3
	Application Content Support	4
	Supported Structured Data Access	4
	Supported Unstructured Data Formats	4
	Oracle Knowledge Services	4
	Infrastructure Services	5
	Application Data Services	5
	Oracle Knowledge Instances	6
	Oracle Knowledge Instance Roles	7
	Communication Between Instances	7
	Transports	8
	Oracle Knowledge Applications	8
	Oracle Knowledge Environments	8
	Environment Roles	9
	Data Synchronization and Propagation	10
	The Inqira Common Environment (ICE)	10
	Language and Localization Support	12
Chapter 2	Configuring Content Acquisition	14
	The Content Acquisition Process	14
	Content Acquisition Components	15
	Accessing Unstructured Documents	15

Configuring Content Acquisition from Web Servers	16
Specifying General Web Collection Settings	16
Specifying Web Collection Starting Points	18
Initializing Web Collections	19
Specifying Advanced Web Collection Settings	20
Configuring Content Acquisition from File Systems	21
Specifying General File Collection Settings	22
Specifying Advanced File Collection Settings	24
Configuring Content Acquisition from Information Manager Repositories	24
Specifying General Information Manager Collection Settings	25
Specifying Publishing Status for an Information Manager Collection	27
Specifying Advanced Information Manager Collection Settings	29
Configuring Content Acquisition from Information Manager Discussion Forums	31
Specifying Discussion Board Collection Parameters	31
Specifying Advanced Discussion Board Collection Parameters	33
Specifying Document Patterns	33
Specify Global Collection Settings	34
Validating Web, File, Information Manager Collections, and IM Discussions	34
Testing Web, File, Information Manager Collections and IM Discussions	35
Viewing Document Details	36
Configuring Content Acquisition from Databases	37
Specifying Database Crawler Parameters	38
Creating Custom Content Collections	39
Configuring a Collection to Use a Custom Crawler	40
Specifying Advanced Collection Settings	40
Configuring Document Attributes	40
Document Attribute Selectors	40
Specifying Document Attributes	41
Configuring Document Filters	41
Specifying Document Filters	41
Configuring Document Supertitles	41
Document Supertitle Selectors	42
Specifying Document Supertitles	42
Specifying Document Conditions	42
Specifying Filter Items	43
Specifying Comparators	43
Specifying Comparison Values	43
Accessing Structured Content	44
Translating User Requests into Queries	44
Querying the Data Source	44
Restricting Query Results	44
Displaying Structured Results	45
Configuring Database Connections for Structured Content	45
Configuring Relational Database Sources	45
Creating a Database View	45
Viewing Configured Schema	45
Viewing Schema Details	46
Schema Field Definitions	46
Connecting to the Data Source	46
Specifying Data Source Connection Parameters	47
Configuring XML Data Sources	47

Associating a Data Source with a Configured Database Crawler Collection	48
Specifying Global Structured Data Retrieval Parameters	48
Setting the Maximum Number of Structured Answers	48
Configuring IML Processing for Structured Queries	49
Specifying XML Translation Scripts	49
Specifying Sub-Query Processing	49
Specifying the XMLTransport Interface	50
Creating Custom Structured Data Source Connectors	50
The Structured Data Retriever Interface	50
Query Set DTD	51
Result Set DTD	51
Working with URL Builders	52
Chapter 3 Translating Content to the Oracle Knowledge Format.	54
Translating Document Structure	54
Access to Document Structure During Request Processing	54
Processing Non-ASCII Characters	55
Supported Data Formats and Converters	55
PDF Document Preprocessing	55
Using the Preprocessor	56
Chapter 4 Operating Oracle Knowledge and Scheduling Jobs	57
Creating and Managing Scheduled Jobs	57
Defining a New Job	59
Specifying Job Tasks and Properties	59
Oracle Knowledge Tasks	60
Single-Collection Content Processing Tasks	60
Cross-Collection Content Processing Tasks	61
Environment Communication Tasks	62
Specifying Collections for a Content Processing Job	63
Scheduling a Job	64
Specifying Job Notification	65
Specifying an Email Host	66
Adding and Modifying Email Notification Groups	67
Adding and Modifying E-mail Group Members	68
Starting a Job Interactively	69
Using Work Clients	70
Defining Tasks for a Work Client	70
Inqira Common Environment Operations	71
Inqira Common Environment Command Operations	71
Viewing Log Data	73
Importing and Exporting Log Files	73
Downloading the Current Log File Set	74
Downloading Log Files for a Specified Event	74
Uploading Log Files	75
Changing the Log Output Level	76
Managing Log Space	76
Additional Parameters to Control Auto Log Purging	77

Chapter 5 Integrating Oracle Knowledge with a Production User Interface . . 79

The Oracle Knowledge Gateway Service	79
Oracle Knowledge Integration Agents	80
Oracle Knowledge Integration Clients	80
The Oracle Knowledge Gateways	80
Using the Gateways	81
Using the SOAP Gateway	81
Deploying the SOAP Gateway	81
Deploying the Installation Override File	81
Configuring the SOAP Communications Options	81
Specifying the URL and Deployment Descriptor	82
Using the Agents	83
Using the J2EE Agent	83
Configuring the SOAP Client for the J2EE Agent	83
Expanding the J2EE Agent Archive	83
Specifying the Gateway URL	84
Deploying the J2EE Agent	84
Accessing the Advanced Configuration Facility and User Interface	85
Using the ASP Agent	85
Prerequisites for Using the ASP Agent	85
Configuring the SOAP Client for the ASP Agent	85
Creating the Application Directory	86
Expanding the ASP Agent Archive	86
Modifying the SOAP Application Variable	86
Adding the New Application to the IIS Configuration	87
Configuring Access from J2EE Clients	91

Chapter 6 Configuring Distributed Applications 92

Implementing Distributed Content Processing	92
The Distributed Content Processing Environment	92
Distributed Content Processing	92
Content Processing Work Distribution	93
Performing Distributed Content Acquisition (Crawl)	93
Performing Distributed Content Standardization (Preprocessing)	93
Performing Distributed Indexing	94
Performing Index Maintenance in a Distributed Content Processing Environment	94
Configuring Distributed Content Processing	94
Configuring the Local Work Client for Distributed Content Processing	94
Configuring Remote Work Clients for Distributed Content Processing	95
The Preprocessing Sub-Collection Creation Process	95
The Indexing Sub-Collection Creation Process	95
Implementing Distributed Request Processing	95
Distributed Request Processing Architecture	96
The Query Coordinator	97
Query Worker Instances	98
Content Collection Groups	98
Distributed Request Processing	98
Configuring the Default Distributed Request Processing Implementation	98

Configuring a Query Coordinator	99
Example Query Coordinator Configuration	100
Defining Content Collection Groups	100
Specifying a Collection Group Name	100
Example Collection Group Name Configuration:	101
Specifying Collection Group Members	101
Example Collection Group Member Configuration:	101
Example Collection Group Configuration	101
Configuring Query Worker Instances	101
Assigning Content Collection Groups to Query Workers	102
Starting Query Worker Instances	103
Assigning Query Coordinators to Query Workers	103
About the Search Component Cache	104
Creating Remote Instances for Distributed Applications	104
Installing Oracle Knowledge on the Remote Processor	104
Copying the Inqura Common Environment to the Remote Processor	104
Starting Inqura Common Environment on the Remote Processor	105
Adding an Instance with a Remote Socket Transport Host	105
Specifying Remote Installation	106
Oracle Knowledge Shortcuts in Remote Instances	106
Chapter 7 Moving Data Between Instances	107
Revision Levels	107
The Synchronization Process	107
Synchronizing Instances with Minimal Interruption	108
The Synchronization Data Package	108
Configuring Data Transfer Between Instances	108
The Installation Configuration File	109
Defining Communication for the Sending Instance	109
Defining Communication for Receiving Instances	109
Example: Data Transfer Transport Definitions	110
Defining the Sending Instance	110
Specifying the Sending Instance Name	110
Specifying the Sending Instance Role	112
Specifying the Sending Instance Transport	112
Example: Sending Instance Definition	112
Defining Receiving Instances	113
Specifying the Receiving Instance Name	113
Specifying the Receiving Instance Role	113
Specifying Receiving Instance Transports	114
Example: Receiving Instance Definition	114
Configuring a Query Coordinator Instance for Synchronization with Minimal Interruption	114
Configuring a Runtime Instance for Use with a Query Coordinator Instance	115
Defining the Synchronization Group	115
Distributing the Installation Configuration File	117
Verifying the Data Transfer Configuration	117
The Propagation Process	118
The Propagation Data Package	118

Configuring Propagation	118
Performing Propagation	118
Performing Data Transfer	120
Prerequisites for Initial Synchronization	120
Chapter 8 Implementing Oracle Knowledge User Authorization	121
Defining an LDAP Authentication Domain	121
Specifying LDAP Authentication Domain Properties	123
Specifying Advanced LDAP Authentication Domain Parameters	124
Using Oracle Knowledge Authorization Roles	125
Pre-Defined Permissions	125
Using the Administrator Role	126
Using the Language Administrator Role	126
Using the Language Development Role	126
Creating and Modifying Authorization Roles	127
Chapter 9 Advanced Configuration Facility	128
System	130
Work Client Status	131
Work Client	131
Click-Through	132
Contact Center Advisor	134
Contact Deflection	136
Crawler Settings	137
Custom Configuration	139
E-mail Notification	139
Workbench	140
Information Manager	140
Rules Engine	141
Language Processor Settings	143
Plug-in Registration	144
Preprocess Configuration	144
Instances	147
Communication Settings	147
Gateways	148
Instance Groups	149
Synchronizer Defaults	150
Service Configuration	151
Content Storage	151
Context Variable	153
Default-Subjectmap	154
Excerpt	155
Excerpt Security	156
Index	157
Log	160
Ontology	160
Structured Data	161
Authorization	162

About This Guide

The *Oracle Knowledge Intelligent Search Administration Guide* is intended for systems and application administrators who need to configure an Oracle Knowledge application.

This preface includes information on:

- [In This Guide](#) - The general organization of this guide.
- [Screen and Text Representations](#)
- [References to Web Content](#)
- [Examples of Product Screens and Text](#)

In This Guide

The *Oracle Knowledge Intelligent Search Administration Guide* is divided into the following sections:

Chapter 1, Oracle Knowledge Application Overview	This section provides an overview of the product architecture, components, and the logical and physical environments in which these components operate.
Chapter 2, Configuring Content Acquisition	This section describes how to configure Oracle Knowledge to collect and index content from supported content sources, including unstructured documents and structured data.
Chapter 3, Translating Content to the Oracle Knowledge Format	This section describes preprocessing, which translates the documents collected by the content acquisition process from their native formats to a standard simplified XML format.
Chapter 4, Operating Oracle Knowledge and Scheduling Jobs	This section describes how to operate Oracle Knowledge by defining and scheduling jobs using the scheduler facility of the Advanced Configuration Facility, and using supported Inquira Common Environment commands.
Chapter 5, Integrating Oracle Knowledge with a Production User Interface	This section describes how to configure Oracle Knowledge request processing instances to communicate with the presentation environment, such as an application server, in which the Oracle Knowledge web-based User Interface is deployed.
Chapter 6, Configuring Distributed Applications	This section describes how to configure an Oracle Knowledge instance to distribute content processing and request processing work among multiple configured instances.
Chapter 7, Moving Data Between Instances	This section describes how to transfer application data, such as indexes, dictionaries, and configuration data, between development, staging, and production runtime instances using synchronization and propagation.
Chapter 8, Implementing Oracle Knowledge User Authorization	This section describes how to implement user authorization within your Oracle Knowledge environment:

Screen and Text Representations

The product screens, screen text, and file contents depicted in the documentation are examples. This guide attempts to convey the product's appearance and functionality as accurately as possible. Application screen content is compared for overall accuracy with screen shots in the guide prior to release. Updates are made where necessary. However, the actual product contents and displays might differ from the published examples.

References to Web Content

For your convenience, this guide refers to Uniform Resource Locators (URLs) for resources published on the World Wide Web, when appropriate. We attempt to provide accurate information; however, these resources are controlled by their respective owners and are therefore subject to change at any time.

Examples of Product Screens and Text

The product screens, screen text, and file contents depicted in the documentation are examples. We attempt to convey the product's appearance and functionality as accurately as possible; however, the actual product contents and displays may differ from the published examples.

Operating System Variations in Examples and Procedures

We generally use Linux screen displays and naming conventions in our examples and procedures. We include other operating system-specific procedures or steps as noted in section headings, or within topics, as appropriate.

We present command syntax, program output, and screen displays:

- in Linux format first
- in other Unix-specific variants only when necessary for proper operation or to clarify functional differences
- in Windows format only when necessary for clarity

Oracle Knowledge Application Overview

This section provides an overview of the product architecture, components, and the logical and physical environments in which these components operate.

Oracle Knowledge Components

The Oracle Knowledge architecture is based on components that run as services. When you install and configure Oracle Knowledge, these services communicate with one another by requesting and responding to requests for data.

The Oracle Knowledge installation process installs and configures the various services as default application that you can use to begin working with Oracle Knowledge.

The default Oracle Knowledge application is configured as a set of components that perform the basic application functions.

See [Oracle Knowledge Services on page 4](#) for more information about Oracle Knowledge services.

Oracle Knowledge is installed with the following major components configured for use. Some of these components operate as Oracle Knowledge services, as described in [Oracle Knowledge Services on page 4](#).

- **The Natural Interaction Engine:** The Natural Interaction Engine provides the runtime request-response processing functions.
- **The Content Processor:** The Content Processor makes your site content available to the application by crawling, preparing, and indexing documents.
- **The Semantic Index:** The semantic index encodes semantic and location information for each document within the application content.
- **The Dictionary:** The Dictionary, which contains the hierarchy of semantic relationships that define the words and phrases used in your business environment, and the rules that determine how the application uses content to respond to user requests.
- **The Rules Engine:** The Rules Engine operates at runtime to compare user questions with the set of Rules in the Dictionary. Each Rule that is true for a request generates an action to perform, ultimately resulting in the application presenting the best possible response to the user.

Application Configuration and Management Tools

Oracle Knowledge provides a complete set of application configuration, management, and analysis tools, including:

- The System Manager, which provides content processing, job scheduling, and log viewing facilities
- The Advanced Configuration Facility, which provides access to the application configuration and administration functions, including content processing and user interface configuration, and administration interfaces
- The Oracle Knowledge Language Workbench, which contains the Dictionary Manager, user management, application testing and quality monitoring

Application Content Support

Intelligent Search supports information retrieval from both structured and unstructured content sources. This section describes the prerequisites and requirements for structured and unstructured content access.

Important! The unstructured information retrieval and structured information retrieval functions are separately licensed Oracle Knowledge components. Consult your account representative for more information about licensing Oracle Knowledge information retrieval functionality.

Supported Structured Data Access

Intelligent Search supports structured data access from the following JDBC-compliant databases:

- Microsoft SQL Server
- Oracle

Supported Unstructured Data Formats

Intelligent Search unstructured content processing supports the following document and data formats:

- Adobe PDF
- ASCII text, including Usenet newsgroup articles
- Hypertext Markup Language (HTML)
- Microsoft Office:
 - Excel
 - PowerPoint
 - Word
- Rich Text Format (RTF)
- Extensible Markup Language (XML)

Oracle Knowledge Services

Oracle Knowledge services are long-running processes that respond to requests from other Oracle Knowledge services to send and receive data.

Oracle Knowledge services include:

- *Infrastructure services*, which transport data within and between instances to define the roles and control the behavior of the various Oracle Knowledge components

- *Application data services*, which store Oracle Knowledge application data, such as processed application content and Dictionary data

You can configure Oracle Knowledge services as members of more than one application; however, a service can only operate within one application at a time. See [Oracle Knowledge Applications on page 8](#) for more information.

You can create distributed Oracle Knowledge application by establishing communication between services on separate instances, as described in [Chapter 6, Configuring Distributed Applications](#)

Infrastructure Services

The Oracle Knowledge infrastructure services include:

Service	Description
Configuration Service	Maintains configuration data associated with defined instances.
Scheduler Service	Provides the mechanism to schedule and assign (distribute) scheduled tasks.
Gateway Service	Provides the entry point for requests into the system.
Synchronization Service	Provides the mechanism to update applications with new application data and configuration information within a production environment.

Note: This section describes only the important infrastructure services from an implementation perspective; it is not a complete list of the Oracle Knowledge services.

Application Data Services

The Oracle Knowledge application data services include:

Service	Description
Content Service	Maintains the acquired application content and additional document metadata, including any Oracle Knowledge document attributes assigned during the acquisition process.
Dictionary Service	Maintains the Dictionary Rules and additional objects, such as domains, domain groups, and users.
Excerpt Service	Provides the mechanism to construct answers from unstructured content.
Index Service	Stores the indexes, which contains the data that supports answer retrieval from unstructured content.
Log Service	Provides the mechanism for writing logs.
Ontology Service	Maintains the concepts and other linguistic data used to analyze language within application content and user requests.

Note: This section describes only the important infrastructure services from an implementation perspective; it is not a complete list of the Oracle Knowledge services.

Oracle Knowledge Instances

When you install and configure Oracle Knowledge, the result is a configured application, containing configured Oracle Knowledge services, suitable for general Oracle Knowledge development such as defining and performing content processing and working with the Dictionary Manager.

You can install and configure additional instances to perform specialized functions, such as language development, content processing, testing, and production request processing, and share the resulting application and configuration data in a distributed production environment.

You define specialized instances by specifying an instance role. The default instance is defined as a Scheduler instance. See [Oracle Knowledge Instance Roles on page 7](#) for more information on the various instance roles.

Oracle Knowledge Instance Roles

You can define and configure Oracle Knowledge instances to perform the following supported roles within your environment.

Role	Description
Tools	An instance configured to support application development activities, particularly in language development within environments that support remote updates to a central Dictionary repository.
Scheduler	An instance configured to distribute work to configured work client instances.
Work Client	An instance configured as a client to receive scheduled tasks from a central scheduler instance.
Runtime	An instance configured to answer questions and return search results.
Query Coordinator	An instance configured to receive data during the synchronization process to minimize runtime instances downtime during the update process. Also used as part of a Distributed runtime processing environment.
Query Worker	An instance configured to perform unstructured searches as part of a distributed runtime processing environment.

Instances configured within production environments operate in either content processing or request processing mode, as described in the following table:

Processing Mode	Role
Content Processing	Tools Scheduler Work Client
Request Processing	Runtime Query Coordinator Query Worker

Communication Between Instances

Oracle Knowledge instances communicate with one another using defined transport methods.

Services within a single instance communicate using local transports that are configured by default. Remote services communicate using transports that you configure.

In development and staging environments, you configure communication between instances by deploying one or more services to one or more transport methods that are supported in your network environment.

In production environments, you import one or more transport methods that are supported in your network environment.

Transports

You can use the following Oracle Knowledge transports to communicate between Java runtime and presentation environments:

- local
- SOAP
- EJB
- socket

You configure transports to access a specified set of services. Each transport definition specifies the services that the transport can access. For example, to configure a SOAP transport for front-end integration, specify the Oracle Knowledge gateway as service within the SOAP transport configuration.

Oracle Knowledge Applications

An Oracle Knowledge application contains one or more Oracle Knowledge instances, configured to perform one or more of the basic content processing or request processing functions. The standard installation and configuration process defines a fully functional default standalone application, consisting of a content processing instance and a request processing instance.

You can define distributed Oracle Knowledge environments that share application configuration and data between specialized instances. You can also configure Oracle Knowledge services within a single instance to support multiple applications.

When you start an Oracle Knowledge instance (JVM), the default application is invoked unless otherwise specified. You can invoke a specified application by calling the application when you start the Oracle Knowledge instance.

Oracle Knowledge Environments

Oracle Knowledge architecture supports the following environments in which you can configure and deploy instances defined, as described in [Oracle Knowledge Instance Roles on page 7](#) to support and control the creation and distribution of application data.

Environment	Description
Development	Development environments can contain multiple Tools, Scheduler (default), Work Client, and Runtime instances used for developing and testing application components and data. The default settings in the Inquire Common Environment application configuration program configures a Development (also referred to as Standalone) environment.

Staging	Staging environments are intended as separate testing environments. They support all operations. Staging environments can contain multiple Scheduler (default), Work Client, and Runtime instances used for validating application components and data prior to synchronization with production instances. The only way to get information into the staging environment is via propagation.
Production	<p>Production environments are intended to support scheduled operations and hands-off administration; they support only scheduled indexing operations and question-answering operations.</p> <p>You move data into production environments using the propagation process. The production environment can contain:</p> <ul style="list-style-type: none"> • One or more Runtime instances configured to receive application data from configured Scheduler instances and to communicate with the production application server • One or more Query Coordinator instances for use by the data synchronization process • One or more Query Worker instances for use in distributed request processing environments

Environment Roles

Environment roles allow you to create independent development, staging, and production environments. You use the development environment to change and test various aspects of the application, such as configuration, Dictionary objects, and advanced features. You can then use the propagation process to move the updated application configuration and data to the staging or production environment.

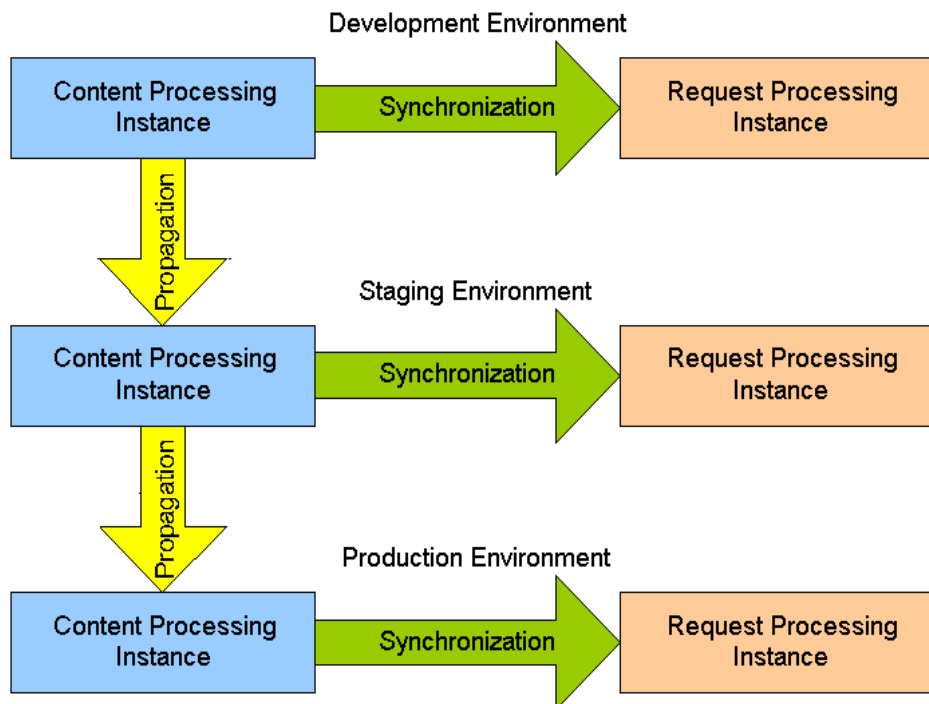
Oracle Knowledge supports the following environment roles, which are applicable to the environments as indicated:

	Tools (content processing)	Indexing (content processing)	Runtime (request processing)
Development	X	X	X
Staging	X	X	X
Production		X	X

Data Synchronization and Propagation

You can move Oracle Knowledge configuration and application data between instances using the following methods:

- *Synchronization* is the method by which you move configuration and application data from a content processing instance to a request processing instance.
- *Propagation* is the method by which you move application configuration and data from development and staging environments to production environments.



Important! Oracle Knowledge does not support synchronization across environments.

See [Chapter 7, Moving Data Between Instances](#) for more information on Synchronization and Propagation.

The Inqira Common Environment (ICE)

The Inqira Common Environment (ICE) is a common operational environment for Oracle Knowledge applications that is installed and configured as part of the standard installation process. The Inqira Common Environment contains tools and utilities that assist in creating, administering, and maintaining Oracle Knowledge instances and applications, enabling you to easily:

- Create and configure applications and instances
- Apply and remove patches and updates to product code
- Build and maintain customized Oracle Knowledge web applications
- Automate operations using external utilities, such as shell scripts

The Inqira Common Environment provides system administration benefits, including:

- A single point of control for managing environment configuration
- Support for implementations using only one instance of Oracle Knowledge product code
- Support for a central Dictionary repository, shared by all applications
- Separate and standard locations for Oracle Knowledge product code and custom code
- Simplified integration and configuration of custom code that uses Oracle Knowledge services
- Access to commonly used environment variables

Language and Localization Support

The following table summarizes language support for Intelligent Search Ontology and Intents.

Language	Ontology and Intent Support						
	Customer Help	Computer	Financial	Insurance	Telecom	Automotive	Utilities Appliances
English	Provided with Installation						
Chinese - Simplified	Provided with Installation						
Chinese - Traditional	Provided with Installation						
French	Provided with Installation						
German	Provided with Installation						
Italian	Provided with Installation						
Japanese	Provided with Installation						
Korean	Provided with Installation						
Portuguese	Provided with Installation						
Russian	Provided with Installation						
Spanish	Provided with Installation						
Arabic	Provided with Installation						
Czech	Provided with Installation						
Dutch	Provided with Installation						
Polish	Provided with Installation						
Slovak	Provided with Installation						
Ukrainian	Provided with Installation						
Danish							
Finnish							
Hebrew	Provided with Installation					Supplied by customer	
Norwegian							
Swedish	Provided with Installation						
Thai							
Turkish	Provided with Installation						
Greek							
Indonesian							
Malaysian							

The following table summarizes the Intelligent Search and Information Manager user interface component localization.

Language	Localized UI		Search	
	Web	Auth	Keyword	NLP
English				
Chinese - Simplified				
Chinese - Traditional				
French				
German				
Italian				
Japanese				
Korean				
Portuguese				
Russian				
Spanish				
Arabic				
Czech				
Dutch				
Polish				
Slovak				
Ukrainian				
Danish				
Finnish				
Hebrew				
Norwegian				
Swedish				
Thai				
Turkish				
Greek				
Indonesian				
Malaysian				

Provided
with Installation

Supplied by
customer

Configuring Content Acquisition

Oracle Knowledge responds to end-user requests by providing answers derived from your organization's content. Oracle Knowledge acquires content from a variety of sources, including unstructured documents and structured data.

Oracle Knowledge acquires unstructured content using its content processing system to collect, translate, and index documents for use by the request and response processing components. Oracle Knowledge acquires structured data by connecting directly to the source database, issuing user requests as SQL queries, and presenting the results.

You configure the content acquisition process using various types of content Collections to access your organization's content. You can configure content acquisition to access content from multiple servers, file systems, databases, and other repositories, as described in [Accessing Unstructured Documents on page 15](#) and [Accessing Structured Content on page 44](#).

You can also configure the application to use both the structured data and unstructured data retrieval modules to return answers from text fields within database content.

Additional elements of the content acquisition system are described in [Content Acquisition Components on page 15](#).

Note: You must have a configured Content Store available to use the content acquisition process. See Configuring the Content Store Data Source [Oracle Knowledge Intelligent Search Installation Guide](#).

The Content Acquisition Process

The content acquisition process is the first step in content processing. Content acquisition makes your organization's documents and data available to the application for request processing. The content acquisition system comprises components that:

- Copy content from configured servers, file systems, and repositories
- Store the content in a compressed form within the Content Store
- Provide content to the Preprocessor, which converts the content to a standardized format for use by other content processing components, as described in [Chapter 3, Translating Content to the Oracle Knowledge Format](#).

Content Acquisition Components

The content acquisition system contains the following components:

Content crawlers	<p>Content crawlers access content stored on various types of servers, file systems, and in other repositories. Content crawlers traverse specified directories or repositories and copy eligible file contents to the Content Store according to scheduling and collection criteria that you specify. Each scheduled crawler task launches an instance of a specific crawler configuration, and each unique crawler configuration defines a document collection.</p> <p>There are several types of content crawlers, each designed to access a particular type of file system, server, or repository, including:</p> <ul style="list-style-type: none"> • Web • File • Information Manager • Database • Siebel Content
Document collections	<p>Document collections are logical entities defined by unique crawler configurations. Each unique crawler configuration defines a document collection.</p>
Document filters	<p>Document filters define inclusion criteria for directories, documents, or files, as described in Configuring Document Filters on page 41.</p>
Document attributes	<p>Document attributes assign metadata that can be used to restrict information retrieval during request processing, as described in Configuring Document Attributes on page 40.</p>
Document supertitles	<p>Document supertitles assign metadata to create logical groups or categories of documents, as described in Configuring Document Supertitles on page 41.</p>

Accessing Unstructured Documents

You can access unstructured content, such as business documents and web pages, by configuring and scheduling one or more Collections. You can differentiate between groups of documents in a single format by defining multiple Collections; each Collection is a unique configuration.

Note: A single document can belong to only one Collection.

Oracle Knowledge provides the following Collection types to access unstructured content:

Web	<p>Accesses content stored on HTTP (web) servers. You configure Web Collections using the System Manager, as described in Configuring Content Acquisition from Web Servers on page 16.</p>
File	<p>Accesses content stored on file systems. You configure File System Collections using the System Manager, as described in Configuring Content Acquisition from File Systems on page 21.</p>

Information Manager	Accesses content stored within an Information Manager repository. You configure System Manager Collections using the System Manager, as described in Configuring Content Acquisition from Information Manager Repositories on page 24 .
Information Manager Discussion Board	Accesses content stored within an Information Manager repository. You configure System Manager Collections using the System Manager, as described in Configuring Content Acquisition from Databases on page 37 .
Database	Accesses unstructured content stored within databases. You configure Database Collections using the Advanced Configuration Facility, as described in Configuring Content Acquisition from Databases on page 37 . Note: See Accessing Structured Content on page 44 for information on accessing structured content from databases.
Siebel Application content	Accesses unstructured content and associated metadata stored within Siebel content repositories. You configure Database Collections using the Advanced Configuration Facility. See the <i>Intelligent Search Siebel Integration Guide</i> , or contact your Oracle Knowledge account representative for more information.

Configuring Content Acquisition from Web Servers

You configure content acquisition from web (HTTP) servers by configuring and scheduling one or more Web Collections to access the desired directories and copy new or modified content into the application.

You can configure multiple Web Collections to access documents located on different servers, or that have different collection requirements.

You define Web Collections using the System Manager's collection definition screens to:

- Specify general Collection information, such as the location of the content and rules to qualify and disqualify documents, as described in [Specifying General Web Collection Settings on page 16](#)
- Specify rules to set acceptance criteria for the initial Collection and subsequent updates, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- Test the Collection, as described in [Testing Web, File, Information Manager Collections and IM Discussions on page 35](#).

Note: You cannot configure Web Collections to access other URL types, such as FTP. Instead, configure a custom crawler, as described in [Creating Custom Content Collections on page 39](#).

You use a Web Collection by scheduling a job, as described in [Chapter 4, Operating Oracle Knowledge and Scheduling Jobs](#). The Web Collection locates content based on specified starting point URLs and other parameters that you specify, as described in [Specifying General Web Collection Settings on page 16](#) and [Configuring Content Acquisition from Information Manager Discussion Forums on page 31](#).

Specifying General Web Collection Settings

You define a Web Collection by specifying the following general information on the System Manager Collection Definition page:

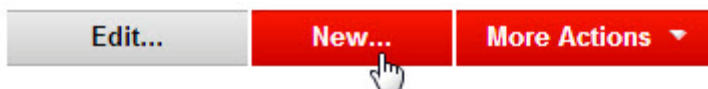
- The name of Collection
- The location of the content that you want to include

- Rules to qualify and disqualify documents for inclusion
- The language and character set of the documents in the Collection
- The method to create the URLs that the application uses to display documents as answers

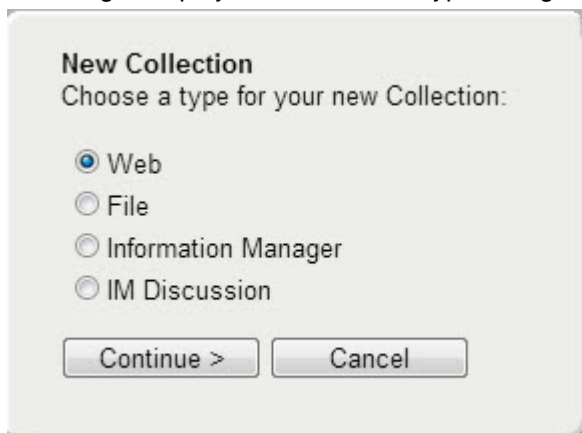
Note: You can also specify additional advanced Collection parameters, such as whether to collect documents located on related hosts, and whether the content acquisition process accepts cookies from crawled sites, as described in [Specifying Advanced Web Collection Settings on page 20](#).

To define a Web Collection:

- 1 Log onto the System Manager and navigate to the Collection List
- 2 Select **New** from the Collection List page:



The System Manager displays the Collection Type dialog:



- 3 Select the **Web** Collection option.
The System Manager displays the Web Collection Definition screen.
- 4 Specify the following Collection settings:

Name	Specify a name for the Collection.
Crawler Starting Points	Specify one or more fully-qualified URLs as starting points for content acquisition, as described in Specifying Web Collection Starting Points on page 18 .

You can specify optional authentication, form field, and cookie information required for site access using the Initialize Crawler option, as described in [Initializing Web Collections on page 19](#).

Accept these Document Patterns	Specify one or more optional document patterns. Enter each pattern on a separate line within the field. Document patterns are regular expressions that logically define desired document characteristics. See Configuring Content Acquisition from Information Manager Discussion Forums on page 31 for more information. Note: The content acquisition process accepts all documents by default; in most cases you do not need to specify explicit document acceptance patterns.
---------------------------------------	---

Reject these Document Patterns (For this Collection Only)	Specify one or more optional document patterns for exclusion from the collection. Enter each pattern on a separate line within the field. Document patterns are regular expressions that logically define desired document characteristics. Note: The content acquisition process uses a set of common reject patterns by default; in most cases you do not need to specify explicit document rejection patterns.
Reject these Document Patterns (For all Collections)	Displays currently specified document exclusion patterns that apply to all collections. Note: The entire list displays only on the Global Collection Settings page. You can specify additional patterns to exclude documents from all collections using the Edit Common Reject Patterns option, as described in Specify Global Collection Settings on page 34 .
Additional Documents	Specify any additional individual documents that would otherwise not be accessed by the crawler configuration. The crawler does not follow links from these documents to other documents.
Language	Select the language used in the collection documents. The default value is en-US, which specifies that the application automatically determines the language and character encoding based on document properties.
Encoding	Select the character encoding used in the collection documents. The default value is AUTO, which specifies that the application automatically determines the language and character encoding based on document properties.

You can specify advanced download and MIME type parameters using the Advanced Settings option, as described in [Specifying Advanced Web Collection Settings on page 20](#).

URL Builder | configure a URL builder to convert the URLs of the crawled content to URLs that can be accessed by end users, if necessary. See [Working with URL Builders on page 52](#) for more information.

- 5 Select **Next >** to continue the process by specifying collection validation settings, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- 6 Select **Cancel** to return to the Collection List without defining a collection.

Specifying Web Collection Starting Points

Web Collection starting points are URLs that define where the web crawling process begins. You can specify multiple starting points for a single crawler.

You specify starting points using the full protocol and path name of the URL, for example, `http://my_company.com`.

Enter each starting point on a separate line within the text field.

No special characters or delimiters are required; however, you can use additional preceding syntax to specify:

- Whether the URL is a jump page, for example:

```
[jump]http://www.mycompany.com/jumppage
```
- A maximum number of levels of URL qualifiers to crawl during content acquisition.

For example:

```
[depth=2]http://www.mycompany.com
```

specifies to crawl only two directory levels below the initial starting point, such that the contents of the directory:

```
http://www.mycompany.com/level_1/level_2/level_3
```

would not be included in the collection. The default crawl depth is 1000.

The web crawler follows and acquires the content linked to each starting point, as permitted or restricted by additional collection definition parameters.

Initializing Web Collections

You can specify various types of information required to access the web content that you are including in the Collection, including user authentication, form data, proxy server information, and cookies.

Authentication	Select an optional authentication method. Valid options are:	
	NTLM	Specify Windows NT LAN Manager protocol
	Basic	Username and password in plain text
	None	Specifies no authentication information
	The fields below display depending on the selected type of authentication:	
	Username	Specify the user name to use when accessing content
	Password	Specify the user name to use when accessing content
	Domain	(Valid only for NTLM) specifies the NTLM domain to which the client belongs
	Realm	Specifies an optional Kerberos realm if required for access in an environment configured to trust non-Windows-brand operating system Kerberos realms
Host	Specify an optional host name of a proxy server	
Port	Specify an optional port to connect to a proxy server	
Form Action	Specify optional document URLs to which you apply form field values to use as input for required form fields within crawled pages	
	Note: The System Manager automatically deletes unused fields and forms when you save the initialization properties.	

Field 1,2,...	<p>Specify the HTML field names and corresponding values to submit for each form and field. Valid values are:</p> <ul style="list-style-type: none"> • text • password • hidden • textarea • checkbox • radio • select/option • submit <p>You can add additional forms and fields to the page using the Add a Field and Add a Form options.</p> <p>For more information on HTML form fields, refer to the World Wide Web Consortium web site: http://www.w3.org.</p>
HTTP Headers	Specify optional key-value pairs (one pair per line) to add to the HTTP header
Cookies	Specify optional cookies

- 1 Enter the properties.
- 2 Select **OK** to save the specified properties and return to the Web Collection definition page.

Important! To permanently save the initialization values, you must complete the definition process and save the collection.
- 3 Select **Cancel** to discard the specified properties and return to the Web Collection definition page.

Specifying Advanced Web Collection Settings

You can specify the following advanced download behavior and MIME type parameters for Web Collections:

Max Download Threads	Specify the maximum number of threads that can be allocated to the acquisition process. The number of threads in use can affect JVM, CPU, and web server usage. The default value is 5.
Ignore Robot Config	Specify whether the crawler ignores <code>robots.txt</code> files and continue crawling as specified by the specified crawler configuration. Valid values are <code>On</code> and <code>Off</code> . <code>On</code> is the default.
Use Cookies	Specify whether the crawler accepts cookies from the site. Valid values are <code>On</code> and <code>Off</code> . <code>On</code> is the default.
Allow Any Hosts	<p>Specify whether to restrict downloaded pages to the host specified as the starting point (<code>Off</code>), or to download referred pages on hosts other than the specified starting point (<code>On</code>).</p> <p>For example, the starting point <code>www.my_company.com</code> may contain a link to <code>www.partner_co.com</code>. If this parameter is <code>On</code>, the crawler downloads the page. If <code>Off</code>, the crawler does not download the page.</p> <p>Valid values are <code>On</code> and <code>Off</code>. <code>Off</code> is the default.</p>

Flexible Host Name	Specify whether to download any URL in the domain, or to discriminate between variants, for example, <code>www.InQuira.com</code> and <code>ww2.InQuira.com</code> . Valid values are <code>On</code> and <code>Off</code> . <code>Off</code> is the default.
Bandwidth Throttle	Specify a value to limit the amount of crawler traffic to and from the server. Specify a value in <code>bytes per second</code> , for example <code>1024</code> . This parameter is optional. The default is <code>0</code> , which specifies no bandwidth limit.
Download Timeout	Specify an interval in seconds after which the crawler stops trying to download a given document. The default value is <code>20</code> , which specifies that the download process times out after 20 seconds.
Document Comparison Method	Select whether the application considers a document to be changed based on comparing: <ul style="list-style-type: none"> • Last Modified Date • Checksum Only The default is <code>Checksum Only</code> .
MIME Type Filter	Specify optional MIME-type filters. You can specify the minimum and maximum size for each type of document using the following syntax: <pre>[[min-size max-size]=value] mime-type/sub-type</pre> For example, you could exclude PDF files larger than 50MB by specifying: <pre>[min-size=52428800] application/pdf</pre>

- 1 Enter the properties.
- 2 Select **OK** to save the specified properties and return to the Web Collection definition page.

Important! To permanently save the advanced collection parameters, you must complete the collection definition process and save the collection.
- 3 Select **Cancel** to discard the specified properties and return to the Web Collection definition page.

Configuring Content Acquisition from File Systems

You configure content acquisition from file servers by configuring and scheduling one or more File Collections to access the desired directories and copy new or modified content into the application.

You can configure multiple File crawlers to access:

- Documents located on multiple file servers
- Documents on a single file server that have different collection requirements

You define File Collections using the System Manager's collection definition screens to:

- Specify general Collection information, such as the location of the content and rules to exclude documents, as described in [Specifying General File Collection Settings on page 22](#).
- Specify rules to set acceptance criteria for the initial Collection and subsequent updates, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- Test the Collection, as described in [Testing Web, File, Information Manager Collections and IM Discussions on page 35](#).

Specifying General File Collection Settings

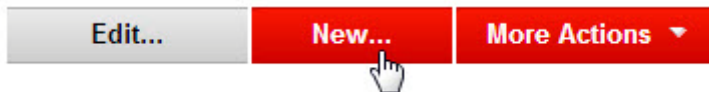
You define a File Collection by specifying the following general information on the System Manager Collection Definition page:

- The name of Collection
- The location of the content that you want to include
- Rules to exclude documents from the Collection
- The language and character set of the documents in the Collection
- The method to create the URLs that the application uses to display documents as answers

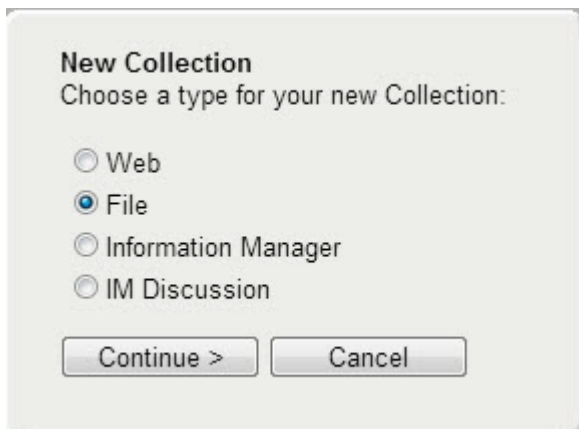
Note: You can specify additional advanced Collection parameters, such as custom file crawler methods and minimum and maximum content size, as described in [Specifying Advanced File Collection Settings on page 24](#).

To define a File Collection:

- 1 Log onto the System Manager and navigate to the Collection List
- 2 Select **New** from the Collection List page:



The System Manager displays the Collection Type dialog:



- 3 Select the **File** Collection option

The System Manager displays the File Collection Definition screen.

- 4 Specify the following Collection settings:

Name	Specify a name for the Collection
Document Directories	Specify one or more directories as starting points for content acquisition. Enter each directory on a separate line within the text field. The File crawler acquires all content within the specified directories and their sub-directories, as permitted or restricted by additional collection definition parameters.

Reject these Document Patterns (For this Collection Only)	Specify one or more optional document patterns for exclusion from the collection. Enter each pattern on a separate line within the field. Document patterns are regular expressions that logically define desired document characteristics. See Configuring Content Acquisition from Information Manager Discussion Forums on page 31 . Note: The content acquisition process uses a set of common reject patterns by default; in most cases you do not need to specify explicit document rejection patterns.
Reject these Document Patterns (For all Collections)	Displays currently specified document exclusion patterns that apply to all collections. Note: The entire list displays only on the Global Collection Settings page. You can specify additional patterns to exclude documents from all collections using the Edit Common Reject Patterns option, as described in Specify Global Collection Settings on page 34 .

You can specify advanced custom crawler and content size parameters using the Advanced Settings option, as described in [Specifying Advanced File Collection Settings on page 24](#).

Language	select the language used in the collection documents. The default value is en-US. Select AUTO to specify that the application automatically determines the language and character encoding based on document properties.
Encoding	select the character encoding used in the collection documents. The default value is AUTO, which specifies that the application automatically determines the language and character encoding based on document properties.
URL Builder	configure a URL builder to convert crawled file system names to display URLs. See Working with URL Builders on page 52 for more information.

- 5 Select **Next >** to continue the process by specifying collection validation settings, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- 6 Select **Cancel** to return to the Collection List without defining a collection.

Specifying Advanced File Collection Settings

You can specify a custom file crawler and control the minimum and maximum content size for the file collection.

Class Name	Specify an optional custom file crawler class, if necessary. See the <i>Intelligent Search Application Developer's Guide</i> for more information on creating a custom file crawler. The default file crawler is <code>com.InQira.content.file.FileCrawler</code> .
Max Size	Specify the maximum size for each document in Kb. The default is 2147483647.
Min Size	Specify the minimum size for each document in Kb. The default is 0.

- 1 Select **OK** to save the specified properties and return to the File Collection definition page.
Important! To permanently save the advanced collection parameters, you must save the collection definition by completing the collection editing process and saving the collection.
- 2 Select **Cancel** to discard the specified properties and return to the File Collection definition page.

Configuring Content Acquisition from Information Manager Repositories

You configure content acquisition from Information Manager repositories by configuring and scheduling one or more Information Manager Collections to access the desired content channels within a repository and copy new or modified content into the application.

You configure multiple Information Manager crawlers to access:

- Documents stored in multiple Information Manager repositories
- Documents assigned to different channels within a repository

The application performs content processing on all eligible Information Manager content records, including documents stored as attachments to content records.

You define Information Manager Collections using the System Manager's collection definition screens to:

- Specify general Collection information, such as the location of the content and rules to exclude documents, as described in [Specifying General Information Manager Collection Settings on page 25](#).

Note: You can also configure the Information Manager Collection to access documents based on publishing status.

- Specify rules to set acceptance criteria for the initial Collection and subsequent updates, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- Test the Collection, as described in [Testing Web, File, Information Manager Collections and IM Discussions on page 35](#).

Specifying General Information Manager Collection Settings

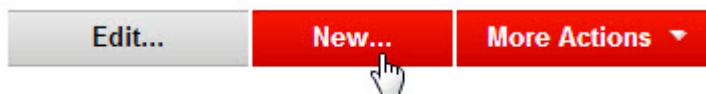
You define an Information Manager Collection by specifying the following general information on the System Manager Collection Definition page:

- The name of Collection
- The Information Manager instance, repository, channel, and publishing status for the content that you want to include
- Rules to disqualify documents from the Collection
- The language of the documents in the Collection
- The method to create the URLs that the application uses to display documents as answers

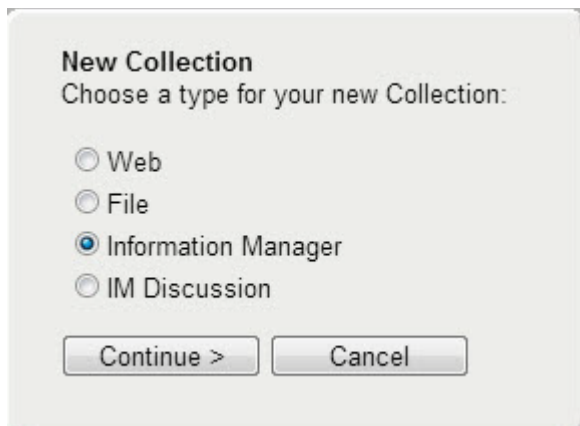
Note: You can also specify additional advanced Collection parameters, such as the number of threads used by the collection process and their priority, and the additional content restriction based on Information Manager content categories, as described in [Specifying Advanced Information Manager Collection Settings on page 29](#).

To define an Information Manager Collection:

- 1 Log onto the System Manager and navigate to the Collection List
- 2 Select **New** from the Collection List page:



The System Manager displays the Collection Type dialog:



- 3 Select the **Information Manager** Collection option

The System Manager displays the Information Manager Collection Definition screen.

- 4 Specify the following Collection settings:

Name	Specify a name for the Collection
IM URL	Specify the URL of the Information Manager application that contains the desired content repository. The System Manager displays the default URL for an Information Manager repository installed and configured within the current instance.

Repository	Specify the unique repository ID (reference key) that contains the desired content. Reference keys are user-assigned identifiers used within the Information Manager application. See A Note On Reference Keys in the Oracle Knowledge Information Manager Administration Guide for more information.	
Channel	Specify the unique Channel ID (reference key) of the Information Manager Channel that contains the desired content Note: You must configure a separate Collection for every Information Manager Channel that you want to include in your application.	
Publishing Status	Specify to acquire published, unpublished, or all latest versions of content Note: See Specifying Publishing Status for an Information Manager Collection on page 27 for more detailed information on acquiring Information Manager content.	
	PUBLISHED	Use this option to acquire only content marked as published within the Information Manager repository. This is the default option.
	ALL	Use this option to acquire: <ul style="list-style-type: none"> published content the latest (in process) draft version of the specified content. <p>This option includes multiple versions of published content records that also have an existing unpublished version (revision) in process.</p>
	IN_PROCESS	Use this option to acquire only the latest version of the specified content records. This option includes both published and unpublished content records, but includes only one (latest) version of any given content record.
Reject these Document Patterns	Specify one or more optional document patterns for exclusion from the collection. Enter each pattern on a separate line within the field. Document patterns are regular expressions that logically define desired document characteristics. See Configuring Content Acquisition from Information Manager Discussion Forums on page 31 for more information. Note: The content acquisition process uses a set of common reject patterns by default; in most cases you do not need to specify explicit document rejection patterns.	
Reject these Document Patterns (For all Collections)	Displays currently specified document exclusion patterns that apply to all collections. You can specify additional patterns to exclude documents from all collections using the Edit Common Reject Patterns option, as described in Specify Global Collection Settings on page 34	

You can specify download control settings and additional content category attributes for the Collection, as described in [Specifying Advanced Information Manager Collection Settings on page 29](#).

Language	Select the language used in the collection documents. The default value is auto. Note: The character encoding for Information Manager content is always UTF-8.
-----------------	--

URL Builder	<p>Configure the Information Manager URL builder (IMURLBuilder) to convert crawled Information Manager repository document paths to display URLs, as described in Working with URL Builders on page 52.</p> <p>You must include:</p> <ul style="list-style-type: none"> • Protocol: Most likely 'HTTP' • Host: The name of your IM server • Port: The port used to access the IM console • Prefix: This should be in the form of: <code><context>/index?page=content&id=</code> This field is used to generate the URL for an answer's click through. <code><context></code> is a place holder which is replaced during the search runtime processing with a webapp instance context, which originates the Search question to Search instance. The value of <code><context></code> can also be a specific webapp instance context (to form a URL that can reach an IM document) <i>if</i> the webapp instance context is <i>fixed</i> (only one such web app instance). • Suffix: <code>&actp=search</code>
--------------------	---

For example, for a host named 'myhost', your URL Builder settings to crawl a channel in the 'Demo' repository, would look like:

Language:	en-US
Encoding:	UTF-8
URL Builder:	IMURLBuilder
* Protocol:	HTTP
* Host:	myhost
Port:	8226
Prefix:	Demo/index?page=content&id=
Suffix:	&actp=search

- 5 Select **Next >** to continue the process by specifying collection validation settings, as described in [Validating Web, File, Information Manager Collections, and IM Discussions on page 34](#).
- 6 Select **Cancel** to return to the Collection List without defining a collection.

Specifying Publishing Status for an Information Manager Collection

You can specify to include documents, (content records) in various publishing states within the acquired content for an Information Manager crawler. Content records within an Information Manager repository can be in the following states:

Status Definition

Live	This is a version of a document that is published and available to all users, who have the security rights to view it. Only one version of a document in a locale can be Live at any one time.
Pending	This is a version of a document that is scheduled to go live at some time in the future, but is not available to users now.
Expired	This is a version of a document that is no longer available to users, because the document version's end date is in the past.
Unpublished	This is a version of a document that is not available to users, because it was never published, or subsequently unpublished as the result of an action performed by a user (see actions below).

The crawler has three options.

Option	Description
Published	Use this option to acquire only the desired content records marked as published within the Information Manager repository. This is the default option.
All	Use this option to acquire both published and the latest (in process) version of the desired content records. This option includes multiple versions of published content records that also have an existing unpublished version (revision) in process.
In_Process	Use this option to acquire the latest version of the desired content records. This option includes both published and unpublished content records, but includes only one (latest) version of any given content record.

Search's IM Crawler handles start date publishing in the following way. One thing to note is if the publishing status is set to In_Process, then it will acquire the latest draft version of a document.

Here is an example with just major versions.

Version	Doc Status	Start Date	End Date	Crawl Publish	Crawl In-Process	Crawl All
1.0	Expired	January 1, 2011	December 31, 2011	No	No	No
2.0	Live	January 1, 2012	December 31, 2012	Yes	No	Yes
3.0	Pending	January 1, 2013	December 31, 2013	No	No	No
4.0	Pending	January 1, 2014	December 31, 2014	No	No	No
5.0	Unpublished	January 1, 2015	December 31, 2015	No	No	No

Here is a simple example showing the impact on In_Process documents.

Version	Doc Status	Start Date	End Date	Crawl Publish	Crawl In-Process	Crawl All
1.0	Expired	January 1, 2011	December 31, 2011	No	No	No
1.1	Draft	January 1, 2011	December 31, 2011	No	No	No
2.0	Live	January 1, 2012	December 31, 2012	Yes	No	Yes
2.1	Draft	January 1, 2012	December 31, 2012	No	Yes	Yes

Here is a more complex example showing the impact on In_Process documents.

Version	Doc Status	Start Date	End Date	Crawl Publish	Crawl In-Process	Crawl All
1.0	Expired	January 1, 2011	December 31, 2011	No	No	No
1.1	Draft	January 1, 2011	December 31, 2011	No	No	No
2.0	Live	January 1, 2012	December 31, 2012	Yes	No	Yes
2.1	Draft	January 1, 2012	December 31, 2012	No	No	No
3.0	Pending	January 1, 2013	December 31, 2013	No	No	No
3.1	Draft	January 1, 2013	December 31, 2013	No	No	No

Specifying Advanced Information Manager Collection Settings

You can specify advanced parameters for an Information Manager Collection to control the number of threads used by the collection process and their priority, and the additional content restriction based on Information Manager content categories.

ORACLE System Manager 8.5.0.0

LOCATION: Collection List → Information Manager Collection → Advanced Settings

Advanced Settings: im_glossary *Indicates Required Field

Download Threads:

Download Priority Adjustment:

Category Attributes:
(One per line)

Document Filter:

Available		Selected
_textFileCollection_max_size	→	
_textFileCollection_min_size	→	
no_empty_files	→	
	←	
	←	

Public User Group:

WARNING: You need full crawl this collection if you change the value of Public User Groups.

Force HTTP Access On Off

Generate Legacy Grants On Off

Forced public: On Off

WARNING: You need full crawl this collection if you change the value of Forced public.

Allow IM PreFilter: On Off

Property	Description
Download Threads	<p>Specify the number of threads to use when acquiring content from the specified repository. The default value is 2.</p> <p>Note: Consider the capacity of the Information Manager database and server when allocating threads to this process.</p>
Download Priority Adjustment	<p>Specify the priority that the Information Manager crawler process threads uses.</p>
Category Attributes	<p>Specify additional Information Manager content category attributes to restrict the collection.</p> <p>Note: See Chapter 4, Content Categories in the <i>Information Manager Administration Guide</i> for more information about content category attributes.</p>
Document Filter	<p>Use the document filters to further restrict the collection.</p>
Public User Group	<p>Specify the reference key of the Information Manager user group that permits public access, if required for content access on the Intelligent Search answer page. Incorrect configuration could make <i>internal</i> content available to <i>external</i> users when searching.</p> <p>If modified, you must run a full crawl to push the change out to the runtime environments.</p> <p>Note: See Chapter 6, Managing Users in the <i>Information Manager Administration Guide</i> for more information about user group reference keys.</p>
Force HTTP Access	<p>Specify whether to access content using only HTTP. If the Information Manager repository and the content processor reside on the same processor, the content processor attempts to access certain data via the file system rather than HTTP by default, to reduce overhead. This option specifies that only HTTP access be used, which may be helpful in diagnosing content processing issues.</p>
Generate Legacy Grants	<p>Determines how the Information Manager crawler sets up the <grant> data in the IQXML for the crawled IM articles. If set to On (True), then the grant value will be based on the IM user group display value (e.g. Employee). If set to Off (False), then the grant value will be based on the IM user group reference key value (e.g. EMPLOYEE). The default value for this setting is Off.</p> <p>Note: Since facets are case sensitive, this setting could cause no results being returned from this IM collection because of a facet restriction not matching the facets for the data. The grant data in the IQXML is then used to generate the CMS_SECURITY.grant facet data.</p>
Forced public	<p>Specifies whether security of all Information Manager documents should be changed to public or not. If set to ON, then all the Information Manager documents are set to be public and visible by all. If set to OFF then the Information Manager security model is used. The default value for this setting is Off.</p> <p>Important! If this is set to ON, a Full Crawl is necessary.</p>
Allow IM Prefilter	<p>Allows for any filtering set up in Information Manager to be used. The default value for this setting is Off.</p>

Property (Continued)	Description (Continued)
OK	Click to save the specified properties and return to the Information Manager Collection definition page.
Cancel	Click to discard the specified properties.

Important! To permanently save the advanced collection parameters, you must complete the collection editing process and save the collection.

Configuring Content Acquisition from Information Manager Discussion Forums

You access unstructured content within Information Manager discussion forums by configuring and scheduling a collection to access the Information Manager repository and copy new or modified content into the application.

You acquire content from multiple discussion boards by defining a unique collection for each discussion board.

Specifying Discussion Board Collection Parameters

You configure content acquisition from Information Manager discussion boards by configuring and scheduling one or more Information Manager discussion board collections to access the desired content channels within a repository and copy new or modified content into the application.

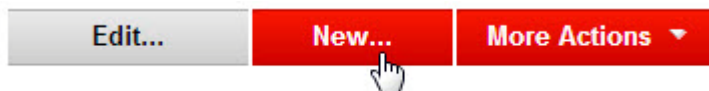
Among the information you specify are:

- The name of Collection.
- The Information Manager instance, repository, and discussion board for the content that you want to include.
- Rules to disqualify documents from the Collection.
- The method to create the URLs that the application uses to display documents as answers.

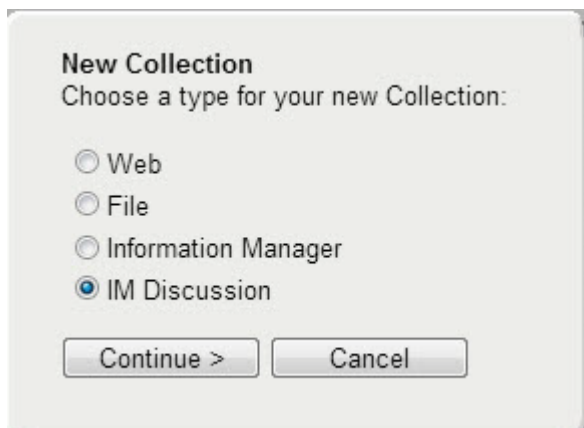
Note: You can also specify additional advanced Collection parameters, such as the number of threads used by the collection process and their priority, and the additional content restriction based on Information Manager content categories, as described in [Specifying Advanced Information Manager Collection Settings on page 29](#).

To define an Information Manager Discussion Board Collection:

- 1 Log onto the System Manager and navigate to the Collection List
- 2 Select **New** from the Collection List page:



The System Manager displays the Collection Type dialog:



3 Select the **IM Discussion** Collection option

The System Manager displays the IM Discussion Collection Definition screen.

4 Specify the following discussion forum configuration items in addition to the common configuration items:

Name	Specify a name for the Discussion Board Collection.
Connection URL	Specify the URL of the Information Manager application that contains the desired discussion board. The System Manager displays the default URL for an Information Manager repository installed and configured within the current instance.
Repository (Key)	Specify the unique repository ID (reference key) that contains the desired discussion board. Reference keys are user-assigned identifiers used within the Information Manager application. See A Note On Reference Keys in the <i>Information Manager Administration Guide</i> for more information.
Discussion Board Ref Key	Specify the unique discussion board ID (reference key). Reference keys are user-assigned identifiers used within the Information Manager application. See Chapter 8, Feedback and Collaboration Features in the <i>Information Manager Administration Guide</i> for more information.
Language	Select the language used in the discussion board. The default value is auto. Note: The character encoding for Information Manager content is always UTF-8.
Document Type Detector	Use this field to specify a custom data type handler, if necessary. Oracle Knowledge is configured to use a default text handler.
URL Builder	Select a URL builder to convert the Information Manager content record ids to URLs that can be displayed as search results. Oracle Knowledge provides a URL builder, <code>com.InQira.content.urlbuilder.IMDiscussionURLBuilder</code> , that you can customize for your application, as described in Working with URL Builders on page 52 .
Protocol	Enter the protocol. For example <code>http</code> .

Host	Enter the host server name.
Port	Enter the port used to connect to the host.
Prefix	Enter a prefix, if necessary.
Suffix	Enter a suffix, if necessary.

You can specify download control settings and additional content category attributes for the Collection, as described in [Specifying Advanced Discussion Board Collection Parameters on page 33](#).

Specifying Advanced Discussion Board Collection Parameters

You can specify the following advanced collection parameters for Information Manager discussion boards:

Parameter	Description
Force HTTP Access	Specify whether to access content using only HTTP. If the Information Manager repository and the content processor reside on the same processor, the content processor attempts to access certain data via the file system rather than HTTP by default, to reduce overhead. This option specifies that only HTTP access be used, which may be helpful in diagnosing content processing issues.
Allow HTML Tags	Specify whether to perform HTML preprocessing within the body of discussion board messages (on), or whether to remove html tags from message preprocessing, which can reduce overhead.
Public User Group (Key)	Specify the reference key of the Information Manager user group that permits public access, if required for content access from the Intelligent Search answer page.
Customize Post	Specify an optional method to customize the output of the discussion board messages, if required.
Forced public	Specifies whether security of all Information Manager documents should be changed to public or not. If set to ON , then all the Information Manager documents are set to be public and visible by all. If set to OFF then the Information Manager security model is used. The default value for this setting is Off .
Reject these url patterns	Specify document patterns to reject one per line.

Specifying Document Patterns

You specify which documents to download from locations within the scope of the content acquisition process by specifying document patterns. For example, you can specify a pattern to exclude image files from downloading.

Document patterns include and exclude documents based on the standard regular expression syntax supported by the Java regular expression standard, `java.util.regex`. See the Java Technology Home Page at <http://java.sun.com> for more information.

The content acquisition process uses the specified patterns to determine download status for each document that it locates. It uses a last-match processing rule; the last regular expression rule that applies to a given

URL determines its download status. The crawler wildcard-matches around the pattern included in the expression by default. For example, the expression:

```
www\InQuira\.com
```

matches any URL that contains the string `www.InQuira.com`, even a query string; however, the expressions

```
http://www\InQuira\.com
```

```
https://www\InQuira\.com
```

restrict the match to the specified domains.

Note: The expression `\.` escapes the period (`.`) character, which otherwise matches exactly one character.

Specify Global Collection Settings

You can specify criteria to exclude documents from all defined collections in the application. The System Manager is configured by default to exclude common graphics and compressed archive formats. You can edit or delete existing document patterns.

Important! Global collections settings apply to all Web, File, and Information Manager collections. You cannot exempt any collections of these types from any global settings. Changes to global settings apply to existing collections. Existing collections may change when you update them after adding or removing patterns from the global list.

To specify global document exclusion patterns:

Reject these Document Patterns

Specify one or more optional document patterns. Enter each pattern on a separate line within the field. Document patterns are regular expressions that logically define desired document characteristics. You can specify any valid regular expressions to create a pattern.

- 1 Select **Save** to save the specified global criteria and return to the collection definition page
- 2 Select **Cancel** to discard the specified global criteria and return to the collection definition page

Validating Web, File, Information Manager Collections, and IM Discussions

You can specify optional acceptance criteria that the application uses to validate the Collection. You can use the validation rules to ensure that the Collection conforms to the expected size, number of new and deleted documents and scope based on specific URL patterns.

The application compares the specified threshold values to a baseline that is established for each collection.

The baseline is set by the first accepted crawl. You can reset the baseline using a pre-defined task, Content Reset, in the administration scheduler, as described in Content Acquisition Tasks.

The application accepts as valid, only content processing results that meet the specified acceptance criteria.

Note: The validation rules are intended to assist in automating content processing; you may need to adjust these rules after initially performing content processing.

You specify validation criteria using the validation, operator and value fields to specify one or more Boolean statements as conditions that the content processing job must meet.

[Validation Field]	Select one of the available criteria: <ul style="list-style-type: none"> • Number of documents • Total Collection Size (min, max in MB) • Presence of documents fitting a URL pattern
[Validation Operator]	Select an operator as the basis of comparing the selected criterion to the actual value in the candidate collection. Valid operators are: <ul style="list-style-type: none"> • > • >= • < • <= • != • contains • does not contain
[Validation Value]	Specify a value for the criterion.

For example, enter the following to specify that the Collection:

- Must contain at least 500 documents.
- Cannot be larger than 1 GB.
- Cannot contain documents with a given URL pattern.

# of docs	>=	500
Collection Size	<=	1000
URL Patterns	does not contain	http://abc.com

- 1 Select **<Back** to preserve the specified validation criteria and return to the Collection definition page.
- 2 Select **Save>** to save the collection definition and validation criteria and proceed to testing the Collection, as described in [Testing Web, File, Information Manager Collections and IM Discussions on page 35](#).
- 3 Select **Cancel** to discard the specified properties and return to the Collection List.

Testing Web, File, Information Manager Collections and IM Discussions

You can test the collection definition by processing the first ten documents associated with each specified collection starting point. The System Manager tests the following aspects of the collection definition:

- Whether the starting points are valid
- Whether the application has proper authentication to access
- Whether the specified URL build method results in valid URLs

To test a collection:

1 Select **Go**



The System Manager displays the progress of the test, and the results:

```

Starting Point: http://techpubs.inquiria.com/
1K View http://techpubs.inquiria.com/Javadoc/index.html
707K View http://techpubs.inquiria.com/InQuira%20Data%20Dictionary/PDF/
2K View http://techpubs.inquiria.com/InQuira%20Data%20Dictionary/HTML
1236K View http://techpubs.inquiria.com/Using%20Advanced%20_ProdName_%20
20_ProdName_%20Features.pdf
2K View http://techpubs.inquiria.com/Using%20Advanced%20_ProdName_%20
1238K View http://techpubs.inquiria.com/Intelligent%20Search%20Configure
20Guide/PDF/Intelligent%20Search%20Configuration%20and%20Adm
2K View http://techpubs.inquiria.com/Intelligent%20Search%20Configure
20Guide/HTML/index.htm
243K View http://techpubs.inquiria.com/IS%20Dev%20Guide%20and%20API%20I
20Ref.pdf
4K View http://techpubs.inquiria.com/IS%20Dev%20Guide%20and%20API%20I

```

You can view the contents of individual documents within each test set by using the View option. To view document details:

2 Select the **View** option that corresponds to the desired document:

```

Starting Point: http://techpubs.inquiria.com/
2K View http://techpubs.inquiria.
1K View http://techpubs.inquiria.
2K View http://techpubs.inquiria.

```

The System Manager displays details for the selected document, as described in [Viewing Document Details on page 36](#).

Viewing Document Details

You can view details for a selected document, including relevant system information to aid in diagnosing problems and verifying successful content processing.

Display URL	Displays a hypertext link to the URL that displays to the end-user as the source document. You can select the link to display the source document in a separate browser window.
Source URL	Displays a hypertext link to the source URL for the document. You can select the link to display the source document in a separate browser window. If the source is not a URL, the document source displays in plain text.

Raw Size	Displays the size of the document after content acquisition. You can download the raw document to an externally saved file.
IQXML Size	Displays the size of the document after conversion to the Oracle Knowledge standard format.
Type	Displays the document type detected by the application.
Doc ID	Displays the internal document ID assigned by the application.
Language	Displays the language of the document detected by the application.
Unique ID	Displays the unique document ID assigned by the application.
Encoding	Displays the document encoding detected by the application.
Checksum	Displays the calculated checksum for the document, which the application uses as acceptance criteria for subsequent job executions.
Facets	Lists the navigation categories (facets) assigned to this document, if applicable.

Configuring Content Acquisition from Databases

You access unstructured content within databases by configuring and scheduling one or more database crawlers to access the desired databases and copy new or modified content into the application. Oracle Knowledge provides a single content crawler type to access two types of database content:

- Documents stored in binary format within a database, such that each record contains one document
- Structured data, such as product catalogs, where each row of collected data is stored as a sentence in a single indexed document

You can configure multiple database crawlers to define various collections of database content. Each unique crawler configuration defines a collection, and each document can belong to only one collection.

Here is an overview of the steps in configuring a database collection:

- 1 Configure a Database Crawler
- 2 Create a Structured Data Service / Specify Schemas for the crawler configured in step 1.
- 3 Under Instances -> Structured Data -> Add a table schema, add a table that is referenced in the database crawler mentioned in step 1.
- 4 Select a Database Collection (Crawler), this should be the same DB collection for which a database crawler is specified.
- 5 Add a key from the table.
- 6 Add other fields that are a part of the database collection (crawler) query.
- 7 Save the configuration,
- 8 Restart the instance to apply these changes.

Note: See [Accessing Unstructured Documents on page 15](#) for information on accessing unstructured content from servers, file systems, and other repositories.

You specify database crawler-specific parameters, as described in [Specifying Database Crawler Parameters on page 38](#).

You can specify additional collection criteria for each collection, as described in [Specifying Advanced Collection Settings on page 40](#).

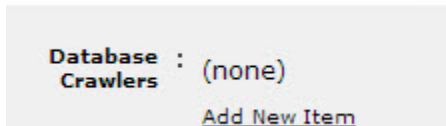
Specifying Database Crawler Parameters

You specify database crawler-specific parameters on the Advanced Configuration Facility Database Crawlers page.

To specify database crawler parameters:

- 1 Select **Crawler Settings** from the Advanced Configuration Facility menu:

The Advanced Configuration Facility displays the Crawler Settings page.



- 2 Select **Edit**, and select **Add New Item** for Database Crawlers:

The Advanced Configuration Facility displays the Database Crawlers page.

ORACLE Advanced Configuration [HELP ?](#)

Editing: Crawler Settings > Database Crawlers

Show Advanced Options **OK** **Cancel**

Item Name ▶ unnamed

Database Crawlers

Available for Unstructured Search : On Off

forcedPublic : On Off

Data Source ▶ SAMPLE_MSSQL [Edit List](#)

Query ▶

Language : Automatic

Encoding : content.encoding[auto]

Document Filter : (none)
[Add New Item](#)

Data Type (default is TXT) : (none)
[Add New Item](#)

Document Attribute Selector : (none)
[Add New Item](#)

Document Supertitle Selector : (none)
[Add New Item](#)

Validation Condition : (none)
[Add New Item](#)

OK **Cancel**

- 3 Specify the following Database crawler-specific configuration items in addition to the common configuration items:

Parameter	Description
Available for Unstructured Search	Select whether or not this content is available for unstructured search. Default is <code>OFF</code> .
forcedPublic	Select whether or not this content is available through the Information Manager UI for public searches. Default is <code>On</code> .
Data Source	Select or define a data source for the crawler.
Query	Specify an SQL query to retrieve relevant records. You can specify queries to retrieve structured data and documents stored within databases. Each record that you collect is stored as a sentence within the Content Store. The query must select the following items in order: <ul style="list-style-type: none"> keys, if required, to access the content the relevant content (columns)
Language	Use this field to specify the language of the content, if necessary. Note: For Information Manager content, the language must be set to something other than AUTO (for example, English (en-US)) or the crawl fails.
Encoding	Use this field to specify the character encoding of the content, if necessary. The default is Automatic, which uses an algorithm to determine the character encoding of the content.
Document Filter	Click Add New Item to use a document filter. Default options are: <ul style="list-style-type: none"> <code>_textFileCollection_max_size</code> <code>_textFileCollection_min_size</code> <code>no_empty_files</code>
Data Type	Use this field to specify a custom data type handler, if necessary. Oracle Knowledge is configured to use a default text handler.
Document Attribute Selector	Click Add New Item to apply an attribute or condition for documents.
Document Supertitle Selector	Click Add New Item to apply a supertitle or condition for documents.
Validation Condition	Click Add New Item to apply a validation condition for documents.

Creating Custom Content Collections

You can define custom Collections to acquire data from (crawl) data sources not supported by the standard content acquisition facility. To create a custom crawler:

- 1 Follow the process described in *Creating a Custom Content Crawler*, as described in the *Intelligent Search Application Developer's Guide*
- 2 Configure a collection to use the custom crawler, as described in [Configuring a Collection to Use a Custom Crawler on page 40](#)

Configuring a Collection to Use a Custom Crawler

You configure a collection to use a custom crawler by specifying:

- 1 The name of the custom crawler configuration class
- 2 Additional collection criteria for each collection, as described in [Specifying Advanced Collection Settings on page 40](#).

Specifying Advanced Collection Settings

You specify additional criteria for each collection by specifying common and crawler-specific configuration items. Specify the following common configuration items:

- The collection name, as described in [Specifying Collections](#).
- Optional document attributes, as described in [Configuring Document Attributes on page 40](#).
- Optional document filters, as described in [Configuring Document Filters on page 41](#).
- Optional document supertitles, as described in [Configuring Document Supertitles on page 41](#).

Specify the database crawler-specific parameters, as described in [Specifying Database Crawler Parameters on page 38](#).

Configuring Document Attributes

Document attributes are metadata that you assign to restrict information retrieval during request processing. You can define document attributes that correspond to content purpose, such as marketing collateral. You specify document attributes for Web, File, and Database crawlers.

You assign document attributes by specifying one or more document attribute selectors in a crawler configuration. Each document attribute selector specifies one or more document attributes and one or more document conditions.

The crawler assigns the specified attributes only to documents for which the specified conditions are true. All specified conditions must be evaluated as true, and all specified attributes are assigned to the document. The assigned attributes are then part of the metadata associated with documents within the Content Store.

Note: You can modify attributes for documents within an existing Content Store by updating attributes in the application configuration and scheduling an attribute update task.

Document Attribute Selectors

Document Attribute Selectors are the mechanism for specifying document attributes. Each document attribute selector specifies:

- One or more document attributes
- One or more document conditions

You can define document attributes that reflect application-specific information retrieval preferences, or configure relevant document attributes defined in an external application, such as a CRM application.

Document conditions are logical statements made up of a filter item, and comparator, and a comparison value, as described in [Specifying Document Conditions on page 42](#).

SPECIFYING DOCUMENT ATTRIBUTES

You can specify one or more document attributes within a document attribute selector. You specify document attributes using the Document Attributes page of the Advanced Configuration Facility. You specify an attribute name and whether the attribute is permanent. Permanent attributes reference document metadata from external sources, for example, an external content management system. Permanent attributes are retained when you use the application scheduler function to update document attributes.

To specify a document attribute:

- Select Document Attribute from the Document Attribute Selector page
- Enter the attribute value. You can specify only one attribute per entry. Attributes can be any alphanumeric string. Spaces and punctuation are not allowed.
- Specify permanent status, if desired, by selecting the Permanent radio button

Configuring Document Filters

You can specify document filters for Collections based on document metadata. Document filters are statements that determine which documents are included in content acquisition.

Document filters specify one or more conditions for inclusion based on document metadata. The crawler tests the conditions against each document's metadata, and accepts only those documents for which all conditions are true.

Specifying Document Filters

You specify Document filters on the Simple Document Filter page of the Advanced Configuration Facility. You can specify any number of Document filters.

Each document filter consists of a name and one or more document conditions.

To specify a Document filter:

- 1 Specify a Document filter name. The name can be any alphanumeric string. Punctuation and spaces are not allowed.
- 2 Specify a Document condition, as described in [Specifying Document Conditions on page 42](#)

Configuring Document Supertitles

Document supertitles are metadata that you assign to create logical semantic groupings of documents. You can specify document supertitles for File, Web, and Database crawlers.

You assign document supertitles by specifying one or more document supertitle selectors in a crawler configuration. Each document supertitle selector specifies one or more document supertitles and one or more document conditions.

The crawler assigns the specified supertitles only to documents for which the specified conditions are true. All specified conditions must be evaluated as true, and all specified supertitles are assigned to the document. The assigned supertitles are then part of the metadata associated with documents within the Content Store.

Note: You can modify supertitles for documents within an existing Content Store by updating supertitle information in the application configuration and scheduling a supertitle update task.

Document Supertitle Selectors

Document Supertitle Selectors are the mechanism for specifying document supertitles. Each document supertitle selector specifies:

- One or more document supertitles
- One or more document conditions

Document supertitles are user-defined values. You can define document supertitles that correspond to semantic categories, for example, product lines or release levels that may not be explicit in the document contents.

Document conditions are logical statements made up of a filter item, and comparator, and a user-defined comparison value. See [Specifying Document Conditions on page 42](#) for a complete description of document conditions.

SPECIFYING DOCUMENT SUPERTITLES

You can specify one or more document supertitles within a document supertitle selector. You specify document supertitles using the Document Supertitles page of the Advanced Configuration Facility.

To specify a Document Supertitle:

- 1 Select Document Supertitle from the Document Supertitle Selector page
- 2 Enter the supertitle value. You can specify only one supertitle per entry. Supertitles can be any alphanumeric string.

Specifying Document Conditions

Document conditions are logical statements that you specify within:

- Document attribute selectors, as described in [Configuring Document Attributes on page 40](#).
- Document filters, as described in [Specifying Document Filters on page 41](#).
- Document supertitle selectors, as described in [Configuring Document Supertitles on page 41](#).

Content crawlers use conditions to determine whether an attribute or supertitle is assigned to a document.

Crawlers determine attribute and supertitle assignments by comparing existing document metadata to all of the conditions specified in the configured document attribute selector or document supertitle selector. Crawlers assign attributes and supertitles only if all specified conditions are true.

Document conditions contain:

- A filter item
- A comparator
- A comparison value

You specify these elements to define a document condition in the form:

`filter_item, comparator, comparison_value`

where:

filter_item	specifies a defined filter item that corresponds to available document metadata. See Specifying Filter Items on page 43 .
--------------------	---

comparator	specifies a Boolean comparator. See Specifying Comparators on page 43 .
comparison_value	specifies a user-defined comparison value. See Specifying Comparison Values on page 43 .

For example, you might set a condition to exclude empty documents by specifying a `document_size` filter item, a comparator of `>` (`greater than`), and a comparison value of `0`.

You define document conditions using the Editing: Content > File Crawlers > Document Attribute Selector > Conditions page of the Advanced Configuration Facility.

Specifying Filter Items

Filter items are the basis of comparison for document conditions, document filters, and file filters within crawler configurations. In document filters and file filters, filter items provide direct matching based on document metadata. In document conditions, filter items are parts of logical statements that also include Boolean comparators and user-defined comparison values.

Filter items consist of:

- A name, which is a simple descriptive label
- A method, which provides the means of accessing and matching the required metadata

The following filter items are available in Oracle Knowledge:

- `getDisplayURL`
- `getDocumentSize`
- `getFetchURL`
- `getLastModificationTime`

Specifying Comparators

Comparators are Boolean operators within document conditions. Comparators specify the basis of comparison between the specified filter item and the user-defined comparison value.

The valid comparators that you can specify within document conditions are:

<code>></code>	<code>greater than</code>
<code><</code>	<code>less than</code>
<code>=</code>	<code>equals</code>
<code>!</code>	<code>not</code>
<code>~</code>	<code>matches a regular expression</code>

You can combine these basic elements to define more complex comparators. For example, you can specify `>=` to indicate `greater than or equal to`.

Specifying Comparison Values

Comparison values are user-defined elements within document conditions. Comparison values specify the value that the document metadata is compared to when the crawler evaluates the document condition. For

example, in a condition to exclude empty documents, the comparison value of 0 specifies the value for document size metadata comparison.

Accessing Structured Content

You can configure Oracle Knowledge to access structured information within databases or in other structured formats, such as Extensible Markup Language (XML). Oracle Knowledge accesses structured data by connecting directly to a configured data source. It does not collect and store structured data for content processing and language analysis, but instead queries the data source directly during request processing.

Note: You can also configure Oracle Knowledge database crawlers to collect unstructured data, such as documents stored within databases or free text within database fields, thereby making that content available for content processing and language analysis. See [Associating a Data Source with a Configured Database Crawler Collection on page 48](#) for more information.

Oracle Knowledge accesses structured information by:

- Translating user requests into structured queries
- Directly querying the data source
- Filtering the query results, if required
- Formatting and presenting the results

Translating User Requests into Queries

The Rules Engine uses Rules within the Dictionary to translate user requests into structured queries. You can configure Rules to produce SQL queries that match the intent of users' natural language questions using the structured query method, as described in [Intelligent Search Language Tuning Guide](#).

Querying the Data Source

Structured query Rules use pre-defined internal schema to produce SQL queries. The structure of the SQL queries is determined by the pre-defined schema.

To enable Oracle Knowledge to generate SQL queries to locate and retrieve the appropriate data, you need to map the structure of your data source to the structure of the pre-defined schema, as described in [Configuring Relational Database Sources on page 45](#).

Restricting Query Results

You can restrict the results of structured queries by including additional IML statements within Rules. The Rules Engine processes IML associated with structured queries by:

- Integrating it into the initial query
- or
- Executing the query, then using the results set as the input for IML processing

You can configure the use of these alternate processing methods to optimize structured query processing for your application by specifying the Maximum IML Results in Query global parameter, as described in [Setting the Maximum Number of Structured Answers on page 48](#).

Displaying Structured Results

The User Interface displays structured results in a tabular format that includes a table heading. You specify the table heading in the Title of Results Rule parameter, as described in the [Intelligent Search Language Tuning Guide](#).

The User Interface also displays a natural language paraphrase of the SQL query. The User Interface uses the following configurable parameters to construct the paraphrase:

- The table description
- The table key
- The field description

as described in [Viewing Schema Details on page 46](#).

Configuring Database Connections for Structured Content

You can configure multiple structured data sources, including relational database tables and XML data repositories. You configure structured information retrieval by:

- Configuring the desired data sources, as described in [Configuring Relational Database Sources on page 45](#) and [Configuring XML Data Sources on page 47](#).
- Setting global parameters for structured information retrieval, as described in [Specifying Global Structured Data Retrieval Parameters on page 48](#).

Configuring Relational Database Sources

The database retriever receives SQL from the Rules Engine and uses the configured internal schema to transform the query into an appropriate query for the configured database. The transformation includes removing unavailable fields and substituting alternate fields, if configured to do so. It executes the query, processes any associated IML, as described in [Configuring IML Processing for Structured Queries on page 49](#), and returns the results.

You configure a relational database by:

- 1 Creating a custom view of your data source, as described in [Creating a Database View on page 45](#).
- 2 Configuring the appropriate application schema to connect to the custom data source, as described in [Associating a Data Source with a Configured Database Crawler Collection on page 48](#).
- 3 Specifying an optional collection for unstructured content processing.

CREATING A DATABASE VIEW

To configure Oracle Knowledge to retrieve answers from your data, you need to create a view that is compatible with the structure of the appropriate internal schema. For information on available internal schema, contact your Oracle account representative.

VIEWING CONFIGURED SCHEMA

Oracle Knowledge is installed with pre-defined schemas. You can view the available schemas, and their defined keys and fields using the Advanced Configuration Facility.

To view the available schemas, select Structured Services from the Advanced Configuration Facility. The Structured Services page lists the installed schemas.

VIEWING SCHEMA DETAILS

To view details for a schema:

- 1 Select Edit on the Structured Services page.
The Editing:Structured Services page displays the schema names as hypertext links.
- 2 Select the desired table name from the list.
The Editing:Structured Services page displays the following fields for the selected table:

Field	Description
Description	A natural language representation of this table that is used in the paraphrase. For example, <code>real-time stocks</code> .
Data Source	The location of the data.
Collection	The crawler configuration defined for acquiring unstructured data used for hybrid retrieval.
Key	The field in this schema that is the key (unique identifier). This is used when creating the paraphrase. For example, <code>stocksymbol</code> .
Field	A list of the fields in the schema. Schema Field Definitions on page 46 describes the parameters that define fields.

SCHEMA FIELD DEFINITIONS

Each field in the schema is defined by the following parameters:

Field	Description
Description	A natural language representation used in paraphrase. For example, <code>book value in dollars</code> .
Exists	Whether or not it exists in this particular database. Possible values are <code>true</code> (default) and <code>false</code> .
Groupable	Not currently implemented.
Alternate	Specifies the field to use if this field does not exist in the data source.

CONNECTING TO THE DATA SOURCE

To configure Oracle Knowledge to retrieve answers from a data source, specify the data source connection parameters, as described in [Specifying Data Source Connection Parameters on page 47](#).

SPECIFYING DATA SOURCE CONNECTION PARAMETERS

The Data Sources page of the Advanced Configuration Facility displays the following data source configuration parameters:

ORACLE Advanced Configuration HELP ?

Editing: Crawler Settings > Database Crawlers > Data Sources > Data Sources

Item Name ▾ Derby

Data Sources

datasourceType ▾ Derby [Edit List](#)

URL ▾

User ▾ Administrator

Password :

Properties

Value : (none)

[Add New Item](#)

OK Cancel

- Specify the following parameters to configure a data source:

Parameter	Description
Item Name	Specifies a required name for the data source
datasourceType	Specifies the database type. Select MS SQLServer or Oracle 9i from the drop-down list, as appropriate.
URL	Specifies the connection URL for the JDBC connection. Enter the value of the connection URL as appropriate for your RDBMS and JDBC driver.
User	Specifies the user name to use for the specified data source
Password	Specifies the password to use for the specified data source
Properties	Specifies any additional required connection properties. Specify the database name in the provided property field. Use the Add New Item option to define any other required database connection properties.

Configuring XML Data Sources

The XML retriever receives SQL from the Rules Engine and uses a configured script to translate the SQL into XML. It sends the XML query to a configured XML transport interface that communicates with the XML data source. It uses an additional script to translate the XML results into a tabular results set for display.

[Query Set DTD on page 51](#) describes the DTD for query scripts. [Result Set DTD on page 51](#) describes the DTD for the results set.

You configure an XML data source by:

- 1 Creating and configuring query and results scripts
- 2 Implementing the XML transport interface
- 3 Specifying an optional collection for unstructured content processing, as described in [Configuring IML Processing for Structured Queries on page 49](#).

Associating a Data Source with a Configured Database Crawler Collection

When you configure a database crawler to collect unstructured data stored within a database, you need to:

- 1 Configure a data source for the collection, as described in [Specifying Data Source Connection Parameters on page 47](#).
- 2 Associate the database crawler collection with the schema defined for the data source

You associate a database collection with a data source using the Structured Services page of the Advanced Configuration Facility. To associate a collection with a schema:

- 1 Select an existing schema, or select **Add New Item**

The Structured Service > Table page displays.

- 2 Select an existing collection, or create a new collection, as described in [Configuring Content Acquisition from Databases on page 37](#).

Specifying Global Structured Data Retrieval Parameters

You specify the following global parameters for structured information retrieval:

- The maximum number of answers to be returned in response to a structured query, as described in [Setting the Maximum Number of Structured Answers on page 48](#).
- The maximum number of IML query results to integrate into a structured query, as described in [Configuring IML Processing for Structured Queries on page 49](#).
- The XML translation scripts to use for retrieving structured information from XML data sources, as described in [Specifying XML Translation Scripts on page 49](#).
- Whether the configured data sources support sub-queries, as described in [Specifying Sub-Query Processing on page 49](#).
- The XML transport interface, as described in [Specifying the XMLTransport Interface on page 50](#).

SETTING THE MAXIMUM NUMBER OF STRUCTURED ANSWERS

You specify the maximum number of answers that structured information retrieval returns for a single query. The default value is 500.

The application retrieves the up to the specified number of results for every query. If a user requests quantities in excess of the specified value, the application retrieves only the maximum number of results.

You specify the maximum number of answers on the Structured Service page of the Advanced Configuration Facility. To specify the maximum number of answers:

- 1 Specify a maximum answers value. Any integer is valid. The default value is 500.
- 2 Select **OK** to save the specified value in your configuration

CONFIGURING IML PROCESSING FOR STRUCTURED QUERIES

The Rules that generate SQL queries can also contain IML to restrict the results of the query. When the IML query is executed, the application:

- Integrates the IML statements into the structured query
- or
- Processes the IML using the results of the structured query as input

A large number of IML results can increase the size of the query such that the database rejects it. The Maximum IML Results in Query specifies the number of results over which the application will post-process the IML results rather than integrate them into the structured query.

You specify the maximum IML results in query on the Structured Service page of the Advanced Configuration Facility. To specify the maximum IML results in query:

- Specify a maximum IML results value. Any integer is valid. The default value is 500
- Select **OK** to save the specified value in your configuration

SPECIFYING XML TRANSLATION SCRIPTS

To configure an XML data source, you need to create scripts to translate:

- Oracle Knowledge SQL queries to client XML queries
- XML results to Oracle Knowledge XML results

[Query Set DTD on page 51](#) and [Result Set DTD on page 51](#) provide document type definitions (DTDs) for the query and results scripts.

You specify query and results scripts for structured retrieval on the Advanced Structured Service page of the Advanced Configuration Facility. To specify the query and results scripts for structured retrieval:

- 1 Select **Advanced** from the drop-down menu on the Edit: Structured Service page
- 2 Enter the URLs for the query and results scripts
- 3 Select **OK** to save the specified value in your configuration

SPECIFYING SUB-QUERY PROCESSING

The SQL queries produced by the application can contain sub-queries; however, some databases do not support sub-queries. This option specifies to execute sub-queries as separate queries.

You specify sub-query processing on the Advanced Structured Service page of the Advanced Configuration Facility. To specify sub-query processing:

- 1 Select **Advanced** from the drop-down menu on the Edit: Structured Service page
- 2 Select the **On** radio button to specify standard sub-query processing
- 3 Select the **Off** radio button to specify that sub-queries be executed as separate queries
- 4 Select **OK** to save the specified value in your configuration

SPECIFYING THE XMLTRANSPORT INTERFACE

Specify a class that implements the XMLTransport interface. This class receives the client's XML query (as a string) and returns the results in the client's XML representation (as a string).

Creating Custom Structured Data Source Connectors

You can create and configure connectors for additional structured data sources. To create a custom connector, you must:

- Implement the `StructuredDataRetriever` interface, as described in [The Structured Data Retriever Interface on page 50](#).
- (Optional) Extend the `StructuredDataAdapter` class which contains helpful code for getting IML results.

You configure the custom connector using the Advanced Structured Service page of the Advanced Configuration Facility. To configure the custom connector:

- 1 Select **Add New Item** under the Connection Type field

The Editing: Structured Service > Connection Type page displays.

- 2 Enter the class name for the custom connector in the Connection field, and select Add
- 3 Select **OK** to save the specified value in your configuration

The Structured Data Retriever Interface

The structured data retriever interface has the following method:

```
public StructuredResult evaluate(Schema schemas,
                               String schemaName,
                               String query,
                               String iml,
                               Datasource data_source,
                               String index,
                               Map options
                               ) throws StructuredException;
```

where:

Argument	Description
schemas	Are the schemas that were configured for the various types of sources (stocks, mutual funds, etc). This is a data structure that allows easy access.
schemaName	Is the name of the schema to use when querying the <code>schemas</code> data structure
query	Is the structured query to execute
iml	Is the restricting IML query that to be run against the unstructured content retrieved from the data source
data_source	Is the data source to which to send the structured query
index	Is the name of the collection that contains the unstructured data to be queried by the IML argument
options	Is a map of the configuration for structured search

Query Set DTD

The query set DTD defines the following elements:

```

<!ELEMENT QUERY ( FIELD*, FCN*, CONST*, SOURCES, QUALIFICATION?, LIMIT?, ORDER? ) >
<!ATTLIST QUERY op(SELECT) >
<!ELEMENT SOURCES (TABLE+) >
<!ELEMENT TABLE (#PCDATA) >
<!ELEMENT LIMIT (START?, QUERY?, COUNT) >
<!ELEMENT ORDER_BY (ORDER+) >
<!ELEMENT ORDER (FIELD) >
<!ATTLIST ORDER dir NMTOKEN #REQUIRED >
<!ELEMENT CONST (#PCDATA) >
<!ELEMENT FCN (ORDER+) >
<!ATTLIST FCN name #REQUIRED >
<!ATTLIST FCN printname #IMPLIED >
<!ELEMENT FIELD (ORDER+) >
<!ATTLIST FIELD name #REQUIRED >
<!ATTLIST FIELD printname #IMPLIED >
<!ELEMENT QUALIFICATION (AND|OR|LIKE|IS_NOT|NE|LT|LTE|GT|GTE|EQ) >
<!ELEMENT AND (AND|OR|LIKE|IS_NOT|IN|NE|LT|LTE|GT|GTE|EQ)+>
<!ELEMENT OR (AND|OR|LIKE|IS_NOT|NE|LT|LTE|GT|GTE|EQ)+>
<!ELEMENT NE ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT EQ ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT LT ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT LTE ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT GTE ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT LIKE ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT IS_NOT ((CONST|FIELD|FUNC), NULL)>
<!ELEMENT NULL>

```

Result Set DTD

The result set DTD defines the following elements:

```

<!ELEMENT const ( #PCDATA ) >
<!ELEMENT field EMPTY >
<!ATTLIST field name NMTOKEN #REQUIRED >
<!ELEMENT headers ( field ) >
<!ELEMENT resultset ( headers, row+ ) >
<!ELEMENT row ( const ) >

```

Working with URL Builders

URL builders create a display URL from the fetch URL used to locate the document for the collection. This URL is then displayed by the search UI.

For example, if the host directory structure and files appeared as:

```
iqbankwest\loan\home\appl1.html
iqbankwest\loan\auto\appl2.html
iqbankwest\loan\home-imp\appl3.html
```

with the server instance running on port 8080; and adding a prefix and a suffix, the parameter settings would be:

```
Protocol: http
Host:: iqbankwest
Port:: 8080
Prefix: help/
Suffix: ?source=IQ
```

then, display URLs would appear as:

```
http:8080//iqbankwest/help/loan/home/appl1.html?source=IQ
http:8080//iqbankwest/help/loan/auto/appl2.html?source=IQ
http:8080//iqbankwest/help/loan/home-imp/appl3.html?source=IQ
```

You can select from several pre-defined URL builders to convert the format of the content locations that can be accessed by content processing to the URLs that web application users can select to access site content as search results:

URL Builder	Collection Type	Parameters
DefaultURLBuilder A placeholder that does not accept any parameters, and does not perform any operation on the crawled content path.	N/A	N/A
RegexReplaceFirstURLBuilder A class that accepts regular expression-based name/value pairs to convert crawled paths to presentation URLs.	Web File	Replace_pattern: enter a regular expression that describes the portion(s) of the crawled path to replace. Replace_value: enter a regular expression that describes desired presentation URL
FileURLBuilder Accepts server directory/filename strings to convert crawled paths to presentation URLs.	File	Protocol: http/https, depending on server environment. Host: hostname or IP address of the http server. Port: port that the http server listens on. Prefix: you want to add to the filename. Suffix: you want to add to the filename.
URLEncodingFileURLBuilder Accepts server directory/filename strings to convert crawled paths to presentation URLs.	File	Protocol: http/https, depending on server environment. Host: hostname or IP address of the http server. Port: port that the http server listens on. Prefix: you want to add to the filename. Suffix: you want to add to the filename.

URL Builder (Continued)	Collection Type	Parameters (Continued)
<p>URLEncodingAbsoluteFileURLBuilder</p> <p>Accepts server directory/filename strings to convert crawled paths to presentation URLs.</p>	File	<p>Protocol: http/https, depending on server environment.</p> <p>Host: hostname or IP address of the http server.</p> <p>Port: port that the http server listens on.</p> <p>Prefix: you want to add to the filename.</p> <p>Suffix: you want to add to the filename.</p>
<p>URLEncodingAbsoluteFileIncludingProtocolURLBuilder</p> <p>Accepts server directory/filename strings to convert crawled paths to presentation URLs.</p>	File	<p>Protocol: http/https, depending on server environment.</p> <p>Host: hostname or IP address of the http server.</p> <p>Port: port that the http server listens on.</p> <p>Prefix: you want to add to the filename.</p> <p>Suffix: you want to add to the filename.</p>
<p>IMURLBuilder</p> <p>Accepts server directory/filename strings to convert crawled paths to presentation URLs.</p>	Information Manager	<p>Protocol: http/https, depending on server environment.</p> <p>Host: hostname or IP address of the http server.</p> <p>Port: port that the http server listens on.</p> <p>Prefix: you want to add to the filename.</p> <p>Suffix: you want to add to the filename.</p>
<p>IMDiscussionURLBuilder</p>	Information Manager Discussion Board	<p>Protocol: http/https, depending on server environment.</p> <p>Host: hostname or IP address of the http server.</p> <p>Port: port that the http server listens on.</p> <p>Prefix: you want to add to the filename.</p> <p>Suffix: you want to add to the filename.</p>

Translating Content to the Oracle Knowledge Format

The Preprocessor translates the documents collected by the content acquisition process from their native formats to a standard simplified XML format.

The Preprocessor contains several component programs that are designed and configured to analyze and translate specific document formats. Supported document formats are described in [Supported Data Formats and Converters on page 55](#).

Note: You must have a configured Content Store to operate the Preprocessor. See [Configuring the Content Store Data Source \[xref IS Install Guide\]](#) for more information.

The Preprocessor produces translated documents that provide semantic information used in processing requests. In its default configuration, the Preprocessor translates all input data without loss, and provides excellent request processing and information retrieval accuracy.

During content processing, the Preprocessor:

- Obtains a list of available documents and their status from the Content Store
- Determines the files to be processed, and their document types
- Translates the files to Oracle Knowledge Input XML
- Stores the Oracle Knowledge Input XML documents in the Content Store

The Oracle Knowledge Input XML format is defined by the Oracle Knowledge Input XML document type definition (DTD).

Note: You must have a configured Content Store to operate the Preprocessor. See [Configuring the Content Store Data Source \[xref IS Install Guide\]](#) for more information.

Translating Document Structure

The Preprocessor retains semantic information encoded in the document structure, such as headings, paragraphs, and lists, when translating documents. The converters translate the semantic elements located within the original documents to analogous elements defined in the Oracle Knowledge Input XML DTD.

Access to Document Structure During Request Processing

During request processing, InQuira Match Language (IML) functions within Dictionary rules can access Oracle Knowledge Input XML semantic structures to optimize request processing. The following IML functions can access Oracle Knowledge Input XML entities:

TITLE	This function specifies to match within document titles.
--------------	--

SUBTITLE	This function specifies to match within document subtitles, including section headings and other named sections of documents.
REFERENCE	This function specifies to match within documents that are linked to by specified terms within link text. See the Intelligent Search Optimization Guide for more information on specifying IML functions within Dictionary rules.

Processing Non-ASCII Characters

The Preprocessor converts all encodings into UTF-8 for processing. All entities are converted to their numeric representation during preprocessing.

The Preprocessor translates Latin 1 – ISO characters (character code 131 – 256) to numeric entity names.

Supported Data Formats and Converters

The Preprocessor translates various document formats to the standard Oracle Knowledge Input XML format.

Document Type	File Extensions
HTML	.htm, .html
XML	.xml
Microsoft Word and Rich Text Format	.doc, .docx, .rtf
Siebel application content, ASCII text, unstructured text within databases	.txt
Portable Document Format (PDF)	.pdf
Microsoft Excel	.xls, .xlsx
Microsoft PowerPoint	.ppt, .pptx

PDF Document Preprocessing

The Preprocessor uses the PDF converter to translate Portable Document Format (PDF) files to Oracle Knowledge Input XML. The PDF converter contains heuristics for determining the following semantic information from the text provided in the PDF file:

- Titles and sections are determined by the PDF title tag; they are also inferred from internal bookmark information, if available
- Paragraphs are determined by sentence proximity
- Additional heuristics distinguish other semantic features, such as tables and multi-column text

Important! The Preprocessor cannot translate encrypted PDF files, because encrypted files cannot be copied. You must use non-encrypted PDF files for Oracle Knowledge content processing.

Using the Preprocessor

You use the Preprocessor by scheduling preprocessing tasks, as described in [Chapter 4, Operating Oracle Knowledge and Scheduling Jobs](#). You can schedule preprocessing tasks as standalone tasks, or as part of a larger set of content processing tasks.

You can schedule preprocessing tasks as standalone tasks, or as part of a content processing task set, as described in [Chapter 4, Operating Oracle Knowledge and Scheduling Jobs](#). The available preprocessing tasks are:

Task	Description
Preprocess Incremental	Use this task to preprocess only documents that have been modified since the last preprocessing task.
Preprocess Full	Use this task to preprocess all documents defined for the specified collection.

Operating Oracle Knowledge and Scheduling Jobs

You can operate Oracle Knowledge by defining and scheduling jobs using the System Manager Scheduler instance. You can use the System Manager Scheduler instance to:

- Schedule pre-defined jobs
Pre-defined jobs consist of commonly used tasks that are grouped together for ease of use.
- Define and schedule customized jobs
Customized jobs are groups of tasks that you define and schedule, either by editing a pre-defined job or by defining a new custom job.

You can schedule jobs to execute once, on demand (at the time they are submitted), or to repeat at specified intervals.

You schedule jobs by:

- Defining the job, as described in [Defining a New Job on page 59](#)
- Specifying tasks and other properties, as described in [Specifying Job Tasks and Properties on page 59](#).

Note: See [Oracle Knowledge Tasks on page 60](#) for a complete description of the tasks that you can schedule.

- Specifying collections (for content processing jobs), as described in [Specifying Collections for a Content Processing Job on page 63](#)
- Specify scheduling parameters, as described in [Scheduling a Job on page 64](#)
- Specifying email notification, as described in [Specifying Job Notification on page 65](#)

Oracle Knowledge executes scheduled tasks using components called work clients. You configure work clients automatically when you use the Inqira Common Environment application configuration process to define local and remote instances and distributed applications. The standard configuration process installs and configures a local work client, as described in [Using Work Clients on page 70](#).

Creating and Managing Scheduled Jobs

You create and manage the scheduled jobs for your application using the Scheduler instance, which is available from the System Manager Job List tab. The Job List displays information about all of the currently scheduled jobs for the application.

Parameter	Description
Job	Displays the job name. The System Manager displays any warning and error icons for the most recent job execution.

Parameter	Description
Status	Displays the state of the Job. Possible states are:
	Complete The execution of the job is complete. Note: The red error icon displays if errors occurred during processing. Select the job and click View... to locate and diagnose processing errors.
	Expired The latest date for which the job was scheduled to execute is past.
	External The job is configured to execute on another processor using a remote workclient. Note: See Creating Remote Instances for Distributed Applications on page 104 for more information on using remote workclients.
	On-Demand The job is configured to execute only interactively from the System Manager.
	Running The job is currently in process.
	Scheduled The job's initial execution is scheduled in the future.
Start Time	Displays the actual time the Job was started.
Next Start Time	Displays the next scheduled start time for the Job.
Env.	Displays the type of application environment in which the Job is configured to run. Possible environments are: <ul style="list-style-type: none"> • Development • Staging • Production Note: A dark blue icon indicates that the Job is configured to execute in that environment. A clear block indicates that it is not.

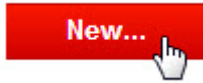
You can:

- Display details about a selected Job, as described in [Viewing Job Status](#).
- Edit a selected Job, as described in [Specifying Job Tasks and Properties on page 59](#) and [Scheduling a Job on page 64](#).
- Define a new job, as described in [Defining a New Job on page 59](#).
- Start a selected job, as described in [Starting a Job Interactively on page 69](#).
- Stop a currently running job using the More Actions Stop Now... option.
- Create a duplicate copy of a selected job using the Duplicate option.
- Delete one or more selected jobs.

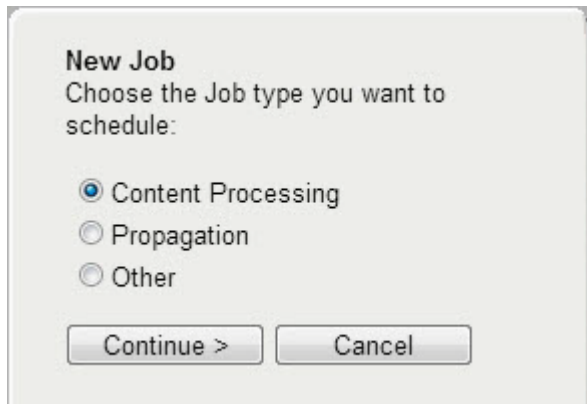
Defining a New Job

To define a new Job:

- 1 Select the **New...** option on the Job List



The System Manager displays the New Job dialog:



- 2 Select the type of job that you want to define:

Content Processing	Define a job to acquire and process content from one or more configured Collections.
Propagation	Define a job to transfer only application data between environments.
Other	Define a custom job.


- 3 Select **Continue >** to specify job tasks and properties, as described in [Specifying Job Tasks and Properties on page 59](#).
- 4 Specify target collections, as described in [Specifying Collections for a Content Processing Job](#).
- 5 Schedule the job, as described in [Scheduling a Job](#).
- 6 Specify notifications, as described in [Specifying Job Notification](#).

Specifying Job Tasks and Properties

To specify the tasks for the job, and additional properties, enter the following information:

Note: The System Manager selects the recommended tasks for the currently selected Job Type by default.

Job Name	Specify a name for the job
-----------------	----------------------------

Valid Environments	<p>This setting makes the job available in one or more of the following environments:</p> <ul style="list-style-type: none"> • Development • Staging • Production <p>This enables you to control Scheduler instance behavior when synchronizing data between development, staging, and production environments.</p>
Job Type	Displays the specified job type.
Job Tasks	<p>The System Manager displays the pre-selected tasks for the selected job type. See the Oracle Knowledge Tasks for descriptions of the default and available tasks.</p> <p>You can remove a task from the job by de-selecting it.</p> <p>You can display additional tasks for inclusion by selecting the Show all tasks option:</p> <div style="text-align: center;">  </div> <p>Note: If you are defining a custom job, select Show all tasks, then select desired tasks. See Oracle Knowledge Tasks on page 60 for a list of all available tasks.</p>

Select **Next >** to continue the process by specifying target collections (for content processing jobs only) and scheduling the Job

Oracle Knowledge Tasks

The following sections describe the available tasks:

- [Single-Collection Content Processing Tasks on page 60](#)
- [Cross-Collection Content Processing Tasks on page 61](#)
- [Environment Communication Tasks on page 62](#)

Single-Collection Content Processing Tasks

Oracle Knowledge provides the following tasks that operate on a per Collection basis. The System Manager pre-selects the appropriate tasks for standard Content Processing:

Task	Description
Status Reset	<p>Use this task to reset status for the specified collection(s). This is the initial step in the standard Content Processing job; you can also use it to recover from failure. For example, you can execute a task to override content acquisition failure due to an exceeded variance threshold for a collection.</p> <p>See Specifying Variance Thresholds for Content Acquisition for more information on setting document count variance thresholds for collections.</p>
Content Reset	Use this task to reset, or delete, the documents within the specified collections. The specified collections can be marked as empty until populated by the next content acquisition process.
Pre-Content Update	Use this task to specify an application-specific custom task required before Content Update.

Task (Continued)	Description (Continued)
Content Update	Use this task to update all documents within the specified collections. The content (documents) defined for the specified collections can be crawled and updated according to the current crawl parameters.
Pre-Document Conversion	Use this task to refer to a custom task specific to your application that should be performed prior to document conversion (preprocessing).
Incremental Document Conversion	Use this task to preprocess only documents that have been modified since the last preprocessing task.
Incremental Document Conversion (Subcoll)	Use this task to create and send sub-collections for incremental preprocessing, which processes only documents that have been added or updated since the last preprocessing operation.
Full Document Conversion	Use this task to preprocess all documents defined for the specified collection.
Full Document Conversion (Subcoll)	Use this task to create and send sub-collections for full preprocessing, which processes all available documents, regardless of their status since the last preprocessing operation.
Pre-Index	Use this task to refer to a custom task specific to your application that should be performed prior to document conversion (preprocessing).
Incremental Index	Use this task to index only documents defined for the specified collections that have been modified since the last indexing task.
Incremental Index (Subcoll)	Use this task to create and send sub-collections for incremental indexing, which processes only documents that have been added or updated since the last preprocessing operation.
Full Index	Use this task to index all documents defined for the specified collection.
Full Index (Subcoll)	Use this task to create and send sub-collections for full indexing, which processes all available documents, regardless of their status since the last preprocessing operation.
Collection Maintenance	Use this task to consolidate the index data (create the reverse index) after the indexing process completes.
Reverse Index Maintenance	Consolidates the reverse index data after the Collection Maintenance task complete.
Meta Index Maintenance	Consolidates the meta index data after the Collection Maintenance task complete.

Note: See [Chapter 6, Configuring Distributed Applications](#) for more information about distributed applications

Cross-Collection Content Processing Tasks

Oracle Knowledge provides the following tasks that operate on all Collections defined for the current job:

Task	Description
Classification	Use this task to create Personalized Navigation taxonomies based on configured content classifiers. See the section on Advanced Features in the Intelligent Search Optimization Guide for more information on the Personalized Navigation feature

Task	Description
Global Maintenance	Use this task to: <ul style="list-style-type: none"> • Create the global data associated with the application indexes • Assign the runtime revision level • Re-deploy the application
Mark Revisions for Synchronization	Use this task to mark current runtime revisions required for synchronization.
Synchronization (Cleanup)	Use this task to create a synchronization package from the current content processing instance, distribute it to the configured request processing instances, and remove the data associated with previous revisions.

Environment Communication Tasks

Oracle Knowledge provides the following tasks related to managing development, staging, and production data:

Task	Description
Pre-Propagation	Use this task to refer to a custom task specific to your application that should be performed prior to propagation.
Propagate to Production	Use this task to create a propagate / move application data from a content processing instance of a development environment to a content processing instance on a production environment. See Chapter 7, Moving Data Between Instances for more information on the propagation process.
Propagate to Staging	Use this task to create a propagate / move application data from a content processing instance of a development environment to a content processing instance on a staging environment. See Chapter 7, Moving Data Between Instances for more information on the propagation process.
Pre-Synchronization	Use this task to refer to a custom task specific to your application that should be performed prior to synchronization.
Synchronization (Cleanup)	Use this task to create a synchronization package from the current content processing instance, distribute it to the configured request processing instances, and remove the data associated with previous revisions. See Chapter 7, Moving Data Between Instances for more information on the synchronization process.
Post-Synch/ Propagation	Use this task to refer to a custom task specific to your application that should be performed after propagation and synchronization.

Specifying Collections for a Content Processing Job

To define a content processing job, you must specify one or more content collections that the job applies to.

All Collections	Select this option to include all configured collections in this job
Specific Collections	Select this option to list the available collections to include in this job. The System Manager updates the page with a list of the currently configured collections: <ul style="list-style-type: none">• Collection: Displays the name of the collection.• Type: Displays the type of collection, for example, HTTP, File, or Information Manager.• Size: Lists the size of the collection.• Number of Docs: Lists the number of documents in the collection.• Associated Jobs: Lists the jobs that process this collection.

Select **Next >** to continue the process by scheduling the Job.

Scheduling a Job

To define a job, you must schedule the date and time for the job to be performed, and the frequency with which it repeats. The System Manager displays the appropriate scheduling options based on your initial frequency selection.

How often do you want to run this Job?	The System Manager provides the following frequency options:	
	On-Demand	Select this option to execute the job interactively
	Once	Select this option to execute the job once at a specified date and time
	Hourly	Select this option to execute the job with these criteria: <ul style="list-style-type: none"> once every specified number of hours beginning at a specified date and time ending at a specified date and time
	Daily	Select this option to execute the job with these criteria: <ul style="list-style-type: none"> once every specified number of days beginning at a specified date and time ending at a specified date and time
	Weekly	Select this option to execute the job with these criteria: <ul style="list-style-type: none"> once every specified day of the week beginning at a specified date and time ending at a specified date and time
	Monthly by Weekday	Select this option to execute the job with these criteria: <ul style="list-style-type: none"> once every specified weekday of the month, for example, every 2nd Wednesday beginning at a specified date and time ending at a specified date and time
	Monthly by Calendar Day	Select this option to execute the job with these criteria: <ul style="list-style-type: none"> once per month at a specified beginning, middle, or final day of the month beginning at a specified date and time ending at a specified date and time
Start Date	Specify the calendar date on which the Job starts.	
Start Time	Specify the time of day at which the Job starts.	
Do not start if wait is longer than ... minutes	Specify an optional interval the System Manager waits before cancelling the job if it is unable to start.	
Repeat	Select the number of times that the job repeats at the specified interval. The System Manager displays additional supporting fields depending on the selection.	
Every (Hourly)	Specify the interval as a number of hours that elapse between occurrences of this job.	
Every (Daily)	Specify the interval as a number of days that elapse between occurrences of this job.	
Every (Weekly)	Select the weekday on which the job is performed.	
Every (Monthly by Weekday)	Select the days of the month on which the Job is repeated.	

Every (Monthly by Calendar Day)	Select the calendar days of the month on which the Job is repeated.
Expiration Date	Select the date on which the Job ends.
Expiration Time	Select the time on the specified day on which the job ends.

- Select **Next >** to continue the process by specifying notification options

Specifying Job Notification

You must specify an email host (sender) and email groups and members (recipients) for the Scheduler instance to use to send job status information.

You configure email hosts, groups, and members using the Advanced Configuration Facility, as described in:

- [Specifying an Email Host on page 66](#)
- [Adding and Modifying Email Notification Groups on page 67](#)
- [Adding and Modifying E-mail Group Members on page 68.](#)

Send e-mail notification to:	Specify the configured email group to send notification of job status
When this Job...	Select one or more events to send notification about. The System Manager selects <code>fails</code> by default. Possible events are: <ul style="list-style-type: none"> • starts • completes successfully • fails

Select **Save** to complete the job definition process

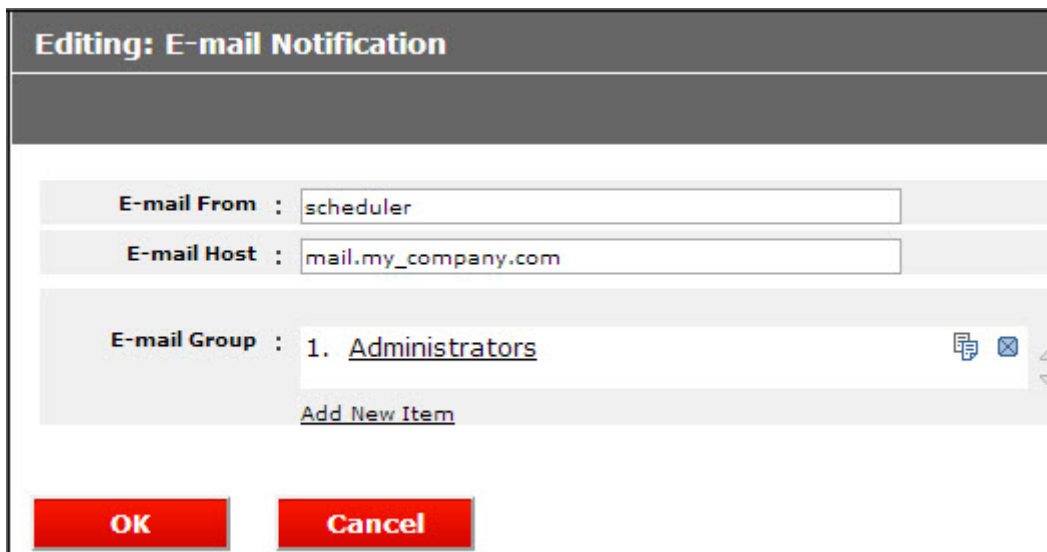
Specifying an Email Host

You must define an email sender for the Scheduler instance email notification feature. You must specify the address of a valid email account on the email server configured for the application.

To specify an email sender for the application:

- 1 Select **Email Notification** from the Advanced Configuration Facility, and select **Edit**

The **E-mail Notification** page displays the **E-mail Host** field.



The screenshot shows a web interface titled "Editing: E-mail Notification". It contains three input fields:

- E-mail From :** scheduler
- E-mail Host :** mail.my_company.com
- E-mail Group :** 1. Administrators (with a dropdown arrow and a plus icon)

Below the "E-mail Group" field is a link labeled "Add New Item". At the bottom of the form are two red buttons: "OK" and "Cancel".

- 2 Enter a valid SMTP server for your environment, for example, `mail.my_company.com`

Note: The SMTP server must be configured to allow relaying for the Oracle Knowledge application. Consult your system administrator for more information.

Adding and Modifying Email Notification Groups

You must use one or more email groups for the email notification feature. Email groups consist of one or more valid email addresses. The Scheduler instance sends notification messages to the addresses within the group when the associated job completes.

Note: The Administrators group is configured by default; however, there are no default email addresses configured within the Administrators group.

To add or modify an email recipient group:

- 1 Select **E-mail Notification** from the Advanced Configuration Facility, and select **Edit**.

The **E-mail Notification** page displays the **E-mail Host** field.

The screenshot shows the configuration page for E-mail Notification. It includes the following fields and controls:

- E-mail From :** scheduler
- E-mail Host :** mail.my_company.com
- E-mail Group :** 1. [Administrators](#)
- [Add New Item](#)
- OK** button
- Cancel** button

- 2 Select an E-mail Group to modify, or select the **Add New Item** link.

The **E-mail Group** page displays.

The screenshot shows the configuration page for E-mail Group. It includes the following fields and controls:

- Editing: E-mail Notification > E-mail Group**
- E-mail Group Name :** Support
- E-mail Addresses :** (none)
- [Add New Item](#)
- OK** button
- Cancel** button

- 3 Specify an **E-mail Group Name**, for example, Support
- 4 Specify group member addresses, as described in [Adding and Modifying E-mail Group Members on page 68](#).

Adding and Modifying E-mail Group Members

To add or modify E-mail group member addresses:

- 1 Select **Email Notification** from the Advanced Configuration Facility, and select **Edit**.

The **E-mail Notification** page displays the **E-mail Host** field.

E-mail From : scheduler

E-mail Host : mail.my_company.com

E-mail Group : 1. Administrators

[Add New Item](#)

- 2 Select an E-mail Group to modify, or select the **Add New Item** link.

The **E-mail Group** page displays the E-mail Group Name field.

E-mail Group Name : Support

E-mail Addresses : (none)

[Add New Item](#)

- 3 Specify an address to modify, or select **Add New Item**.

The E-mail Addresses page displays the Address field.

Editing: E-mail Notification > E-mail Group > E-mail Addresses

Address ▶ bob@my_company.com

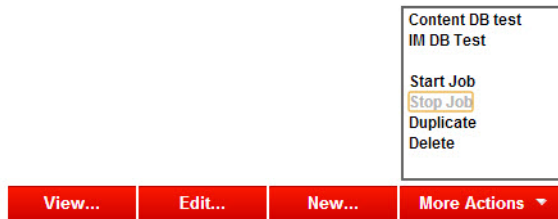
OK **Cancel**

- 4 Enter a valid address.
- 5 Repeat the process to add member addresses as needed.

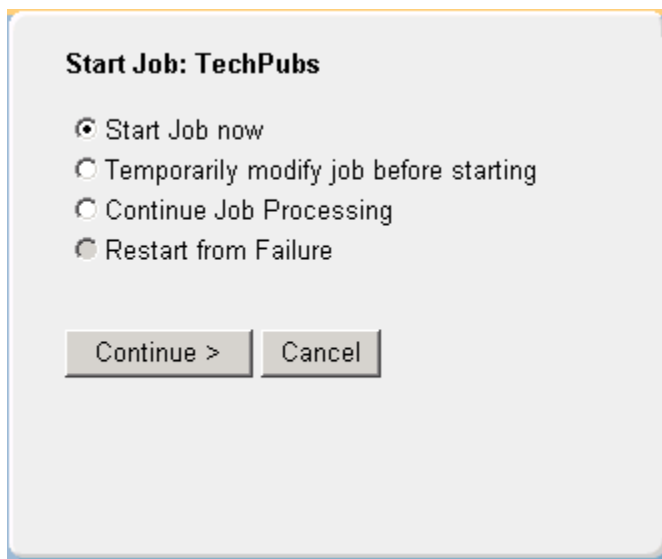
Starting a Job Interactively

To start a configured job interactively from the Job List in System Manager:

- 1 Select the job from the Job List
- 2 Select **More Actions**, click **Start Job...**



The Start Job dialog displays:



- 3 Select an option from the Start Job dialog display

Start Job now	Submit the selected job for immediate processing using its currently saved configuration.
Temporarily modify job before starting	Modify the job configuration only for a single occurrence.
Continue Job Processing	Continue a manually stopped job from the point at which processing stopped.
Restart From Failure	Restart a failed job from the point of failure.

Note: The System Manager includes interactive job information in the job history details, even when the job configuration is temporarily modified.

- 4 Select **Continue >**

The view displays the Job Status screen.

Using Work Clients

Work clients are Oracle Knowledge components that perform the actual work associated with scheduled tasks. The standard Development Application configuration configures a local work client to perform tasks on the local Scheduler instance.

You can define additional instances, each with its own work client, to create a distributed application to reduce processing time for scheduled tasks. Work clients operate as long-running processes that communicate with the Scheduler instance.

The Scheduler instance also provides the mechanism for distributed processing. If a task group contains multiple tasks or collections, the tasks can be distributed to any configured work clients. If there are no work clients, the Scheduler instance itself performs the work.

The work clients register themselves dynamically with the Scheduler instance. The Scheduler instance records the following information for each configured work-client:

- the types of tasks it can perform
- whether it is ready to perform a task

You can limit the tasks that a workclient is allowed to perform, as described in [Defining Tasks for a Work Client on page 70](#).

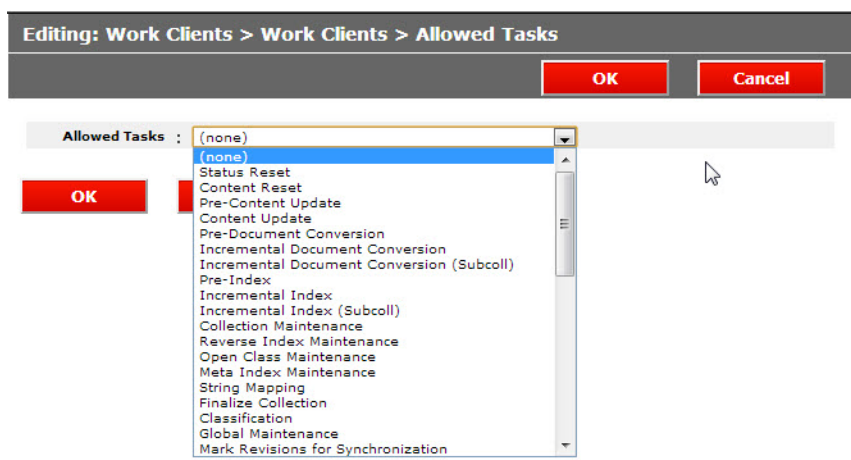
Defining Tasks for a Work Client

You define the allowed tasks for a work client using the Allowed Tasks field on the Work Clients page. The Work Clients page lists the allowed tasks for the currently defined work clients.

To add a task:

- 1 Select the **Add New Item** link in the Allowed Task list for the selected Work Client

The Allowed Task displays the currently allowed tasks for the selected Work Client:



- 2 Select a desired task, then repeat the process for each additional allowed task

Note: See [Oracle Knowledge Tasks on page 60](#) for descriptions of the available tasks.

InQira Common Environment Operations

This section describes the operations that you can perform from the InQira Common Environment command window.

InQira Common Environment Command Operations

The InQira Common Environment provides access to the following scripts and programs:


Program/Script	Description
<code>applyUpdates.bat/sh</code>	Used to apply the OK jar updates for workclients.
<code>batchQualityMonitor.bat/sh</code>	Run the quality monitor in batch mode.
<code>browser.bat/sh</code>	Run the service browser application.
<code>buildWebapp.bat/sh</code>	Use this command to compile the Oracle Knowledge web application for initial use, or to apply customization.
<code>createContentStore.bat/sh</code>	Use this command to create the database tables for the indexed application content.
<code>createQualityMonitorStore.bat/sh</code>	Use this command to create the database tables for the Quality Monitor.
<code>deployApp.bat/sh</code>	Use this command to deploy the Oracle Knowledge web application for initial use, or to apply customization.
<code>indexInfo.bat/sh</code>	Use this command to display information about the Content Store.
<code>printVersion.bat/sh</code>	Use this command to display Oracle Knowledge version information.
<code>tailTheLog.bat/sh</code>	Use this command to display the application log data.
<code>workbench.bat/sh</code>	Use this command to start the Oracle Knowledge Language Workbench application.

Viewing Log Data

You can view and analyze application log data using the Event Viewer.

To access the Event Viewer:

- 1 Select **Tools** from the System Manager.
- 2 Select **Event Viewer**.

Or, select the Event Log icon  located in the top right corner of the Job Status screen.

The Event Viewer reads, categorizes, and displays logged application information by functional area, for example:

- Infrastructure, which contains events related to the application instances and services.
- Scheduler, which contains events related to defining and executing scheduled Jobs.
- Content Acquisition, which contains events related to preprocessing files during a crawl.

Note: The specific categories of event data displayed depend on the configuration, role, and usage of the application.

Within each functional category, the Event Viewer displays a time-based hierarchy that you can expand to locate specific events. You can use the Error (▲), Warning (▲), and Exception (▲) icons to locate nodes within the log that contain these types of events, and drill down to see the relevant details.

The Event Viewer includes functions at the top of displays that you can use to:

- Search for specific text within the entire log file

Note: The Find function supports regular expressions, and is not restricted to the currently selected node.

- Page through the log file
- Save the current log file set to a .jar file for remote analysis
- Save the log files for a specific event for remote analysis

You can restrict the data displayed by the Event Viewer using the Failure, Success, Warning, Review/Exception filters to show one or more of these types of events. You can also change the log output level, as described in the section on [Changing the Log Output Level on page 76](#).

Importing and Exporting Log Files

You can download the entire set of current log files to your local disk drive for further analysis, as described in [Downloading the Current Log File Set on page 74](#). You can also download log files for a specific event, as described in [Downloading Log Files for a Specified Event on page 74](#).

Once downloaded, the log files can then be uploaded and viewed on another instance of the System Manager at another location, as described in [Uploading Log Files on page 75](#).

Important! Log files are stored in the

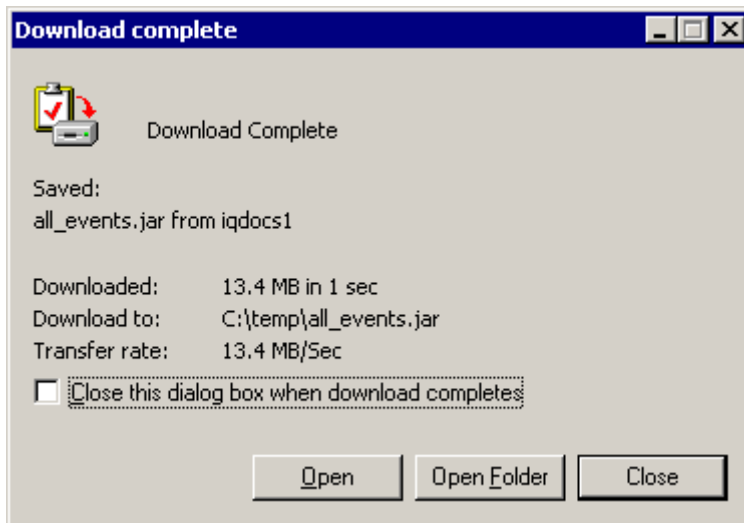
<\$Base>\development\content\data\log\binary\default folder by default. When you upload them, however, you must upload them from the <\$Base>\development\content\data\log\binary_external folder.

Downloading the Current Log File Set

To download the entire set of current log files:

- 1 Select **Current Log Set** as the Logset
- 2 Click **Download** to start the download process
- 3 Click **Save** when prompted

- 4 Select the folder in which to save the .jar file and click **Save**




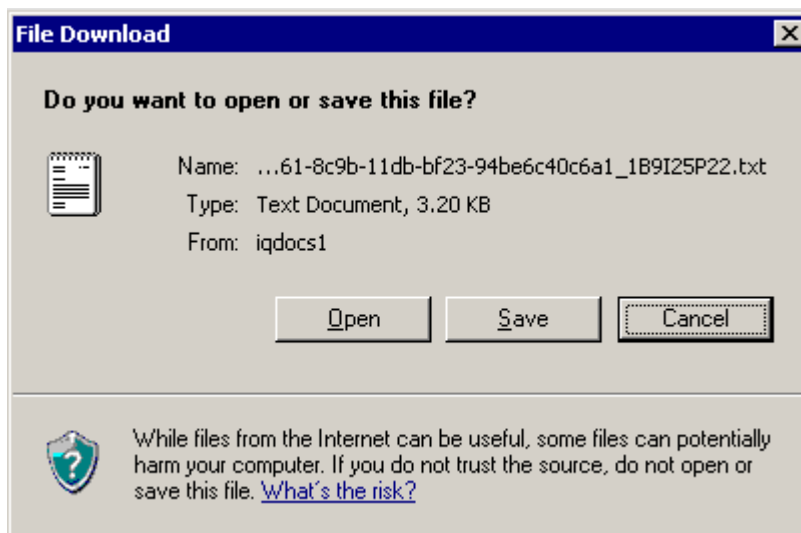
The Event Viewer displays the Download Complete dialog when it has finished saving the .jar file.

- 5 Click **Close** to close the Download Complete dialog

Downloading Log Files for a Specified Event

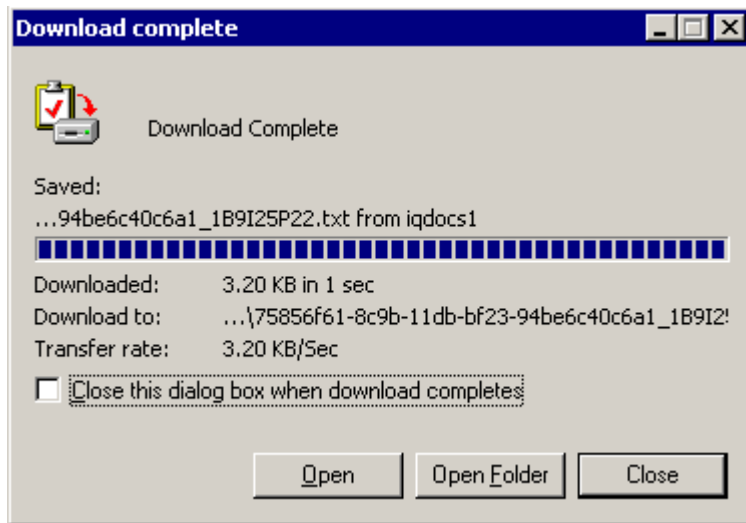
To download a set of log files for a specified event:

- 1 Select **Current Log Set** as the Logset
- 2 Click on the event to download
- 3 Click  at the top of the Log Detail pane to start the download process
The File Download dialog is displayed.



- 4 Click **Save** when prompted

- 5 Select the folder in which to save the .jar file and click **Save**



The Event Viewer displays the Download Complete dialog when it has finished saving the files.

- 6 Click **Close** to close the Download Complete dialog

Uploading Log Files

To upload log files:

- 1 Extract and copy the log files to
`<$Base>\development\content\data\log\binary_external`
- 2 In the Event Viewer, select **External Log Set** as the Logset
- 3 Click **build** to upload the log files

The uploaded log files are displayed in the Event Viewer.

Changing the Log Output Level

You can change the log Intelligent Search log output level to include debug information. To change the log output level to VERBOSE, add the following XML to the `installation.xml` file:

```
<serviceConfiguration name= default
<logService>
<verbosity keyref="choices.verbosity[Debug]" />
</logService>
</serviceConfiguration>
```

To enable output to the console, change the instance definition to include the item in bold below:

```
<instance name="xx"
<bootstrap>
<logStore>
<consoleEnabled>true</consoleEnabled>
....
</logStore>
....
```

```

</bootstrap>
...
</instance>

```

Managing Log Space

Search Scheduler and Runtime instances purge logs automatically.

The configuration for allocating the disk space is defined in the `installation.xml` on the scheduler and each runtime instance and can be found at `$INSTALL_HOME/instances/<INSTANCE_NAME>/installation.xml`.

```

<instance name="default">
  <description>default Instance</description>
  <role keyref="instanceRole[1]" />
  .....
  <logStore>
    <class>com.inquiria.log.LogStoreImpl</class>
    <maxFileSize>5</maxFileSize>
    <maxDirSize>2000</maxDirSize>
    <internalLog>false</internalLog>
    <binaryFileEnabled>true</binaryFileEnabled>
    <consoleEnabled>true</consoleEnabled>
    <textFileEnabled>true</textFileEnabled>
    <maxBlockSize>100</maxBlockSize>
    <maxTextFileSize>100</maxTextFileSize>
  </logStore>
  ...
</instance>

```

The default limit for `maxFileSize` is 5 MB and for `maxDirSize` is 2000 MB. You can configure these sizes as per the disk space.

To change the configuration:

- 1 Stop the Oracle Knowledge Instance.
- 2 Update the `<INSTALL_DIR>\instances\<INSTANCE_NAME>\installation.xml`
- 3 Reconfigure the `<maxFileSize>`, the maximum file size for each log file, to represent the maximum file size in Megabytes (recommended maximum 20M for a runtime instance)
- 4 Also reconfigure the `<maxDirSize>`, the maximum disk space reserved for an instance's log files, to represent the maximum files allowed in one log directory (recommended maximum 2000M for a runtime instance).

Note: If the `<maxFileSize>` is met, yet there is still space in `<maxDirSize>`, a new log file is created and utilized until the `<maxDirSize>` is full

- 1 Restart the instance.
- 2 Repeat on each content processing workclient and runtime instances.

The parameters mentioned above apply to the logs in the `<INSTALL DIR>\instances\<INSTANCE NAME>\<ENV_ROLE>(development, staging, or production)\<INSTANCE_ROLE>(content or runtime)\data\log\binary or text directory` for the instance in which the `installation.xml` file has been modified.

Additional Parameters to Control Auto Log Purging

Log Purging on the scheduler and runtime considers 2 factors:

- a Retention Capacity - The percentage of the `maxDirSize` (configured in the `installation.xml`) of logs to be retained after auto log purging. The default Retention Capacity is 70% of `maxDirSize`.
- b Retention Days - The number of days of logs to retain after auto log purging. The default Retention Days is 14 days.

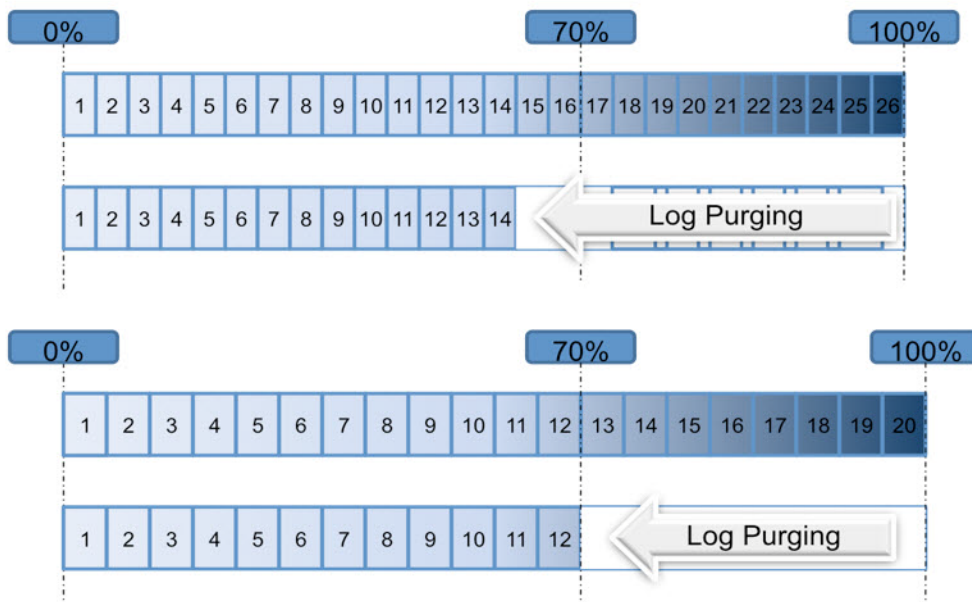
For example, consider 2 scenarios,

- 1 You reach 70% of `maxDirSize` on the 16th day. When the logs reach 100% of the `maxDirSize`, all logs until the 14th day are purged.
- 2 You reach 70% of `maxDirSize` on the 12th day. When the logs reach 100% of the `maxDirSize`, all logs until that day are purged, since the Retention Capacity limit is reached before the Retention Days limit.

In both scenarios, you are guaranteed to have at least 30% free space after Auto Log Purging.

In Scenario 2, fewer days of logs are retained than configured with the Retention Days factor. In order to retain 14 days of logs, `maxDirSize` must be increased.

Log Purging – 2 Scenarios



To change the default capacity and retention days configuration:

- 1 Stop the Oracle Knowledge Instance.
- 2 Update the `#.xml` on that instance located at `<INSTALL_DIR>\instances\<INSTANCE_NAME>\<INSTANCE_ROLE>(development or content)\data\config\#.xml`

```
<task-definition index="47">
  <name>Log history purge</name>
```

```
<shortName>LogPurge</shortName>
<description>purge log history data</description>
  <taskClass>com.inquiria.log.LogHistoryPurgeTask</taskClass>
    <parameters index="0">-r</parameters>
    <parameters index="1">14</parameters>
    <parameters index="2">-c</parameters>
    <parameters index="3">0.7</parameters>
    <parameters index="4">-m</parameters>
    <parameters index="5">>false</parameters>
  <distributed>>false</distributed>
  <needsCollection>>false</needsCollection>
  <subcollection>>false</subcollection>
</task-definition>
```

- 3 Change the parameter indexed 1 to configure the Retention Days.
- 4 Change the parameter indexed 3 to configure the Capacity in decimal. (For example, 60% = 0.6).
- 5 Restart the instance.
- 6 Repeat on each content processing workclient and runtime instance.

For advanced logging for Stellent Server, see [Appendix A: Advanced Stellent Server Logging](#).

Integrating Oracle Knowledge with a Production User Interface

Oracle Knowledge Runtime instances, which process requests and provide responses, must communicate with the presentation environment, such as an application server, in which the Oracle Knowledge web-based User Interface is deployed. The runtime environment communicates with the presentation environment through its Gateway service.

Note: When you perform the default Oracle Knowledge installation and configuration, the Oracle Knowledge runtime communicates with the packaged HTML agent, which is deployed within an Apache Tomcat or WebLogic web server.

The Oracle Knowledge runtime components operate in a Java JVM environment. For production purposes, you can integrate Oracle Knowledge Runtime instance with a web-based client operating in a separate JVM or IIS/ASP environment.

The Oracle Knowledge Gateway Service

The runtime environment communicates with the presentation environment through the Gateway service. The Gateway supports communication through the various agents, clients, and gateways that are provided with the product.

Agents, clients, and gateways are packaged in the product as archives (.jar or .war) that you configure and deploy in your presentation environment.

The following table describes the relevant integration components for various types of presentation environments:

Presentation Environment	Gateway	Agent
Oracle Knowledge web server	N/A (local)	HTML Agent: htmlagent.war/ui.jsp
J2EE web/application server	SOAP	J2EE Agent: j2eeagent.war/ui.jsp
IIS/ASP	SOAP	ASP Agent: aspagent.zip/default.asp

Oracle Knowledge Integration Agents

Oracle Knowledge provides the following agents packaged as archives that you can configure and deploy to support communication between the runtime and presentation environments:

Agent	Archive Name	Description
HTML Agent	htmlagent.war	This agent provides a J2SE compliant web application that performs XSLT processing using the provided integrated (Tomcat) web server.
J2EE Agent	j2eeagent.war	This agent provides a J2EE compliant web application that performs XSL processing using an Oracle Knowledge gateway.
ASP Agent	aspageant.zip	This archive an IIS/ASP-compliant web application that uses SOAP to communicate with the Java runtime environment.

The integration agents are located in the archive directory in the Oracle Knowledge home directory.

Oracle Knowledge Integration Clients

The integration agents use provided clients to access configured gateways. Oracle Knowledge provides the SOAP integration client, which corresponds to the provided gateways:

The integration client is packaged with the integration agent in the archive directory.

The Oracle Knowledge Gateways

Oracle Knowledge standalone gateway supports communication between the runtime environment, which operates in a Java JVM, and Java 1.3 and higher, Microsoft IIS using ASP, and other presentation environments.

Use the following Oracle Knowledge gateway to communicate between runtime and deployed client applications:

Gateway	Description
SOAP	This gateway supports communication between the runtime and J2EE, Microsoft IIS using ASP, and other presentation environments. The SOAP gateway is built and deployed as <code>inquiragw.war</code> from the archive file <code><InQuira_home>/archive/htmlagent.war</code> , after running <code>buildwebapp</code> and <code>deployApp..</code>

You use an Oracle Knowledge gateway by:

- Selecting the appropriate gateway for your presentation environment
- Configuring the gateway, as described in [Using the Gateways on page 81](#)
- Configuring the appropriate agent, as described in [Using the Agents on page 83](#)

Using the Gateways

You must configure the appropriate gateway for the integration agent. Oracle Knowledge provides the SOAP gateway, as described in [Using the SOAP Gateway on page 81](#)

Using the SOAP Gateway

To configure the SOAP gateway:

- Deploy the SOAP gateway archive, as described in [Deploying the SOAP Gateway on page 81](#)
- Deploy the Oracle Knowledge installation override file, as described in [Deploying the Installation Override File on page 81](#)
- Configure the communication options, as described in [Configuring the SOAP Communications Options on page 81](#)
- Deploy the modified archive as appropriate for your application server

Note: The SOAP gateway archive is similar to the HTML Gateway archive, but does not contain the files associated with XSL formatting. The Advanced Configuration Facility is not available in this environment; you need to configure the application by manually editing the `installation.xml` file.

Deploying the SOAP Gateway

To deploy the SOAP Gateway:

- in ICE, run `buildWebApp` and `deployApp`. This deploys `inquiragw.war` to the runtime instance.
or
- place it in the appropriate location for your application server and allow it to be expanded

DEPLOYING THE INSTALLATION OVERRIDE FILE

The installation override file contains installation-specific information required by the application server. It is located at:

```
<InQuira_home>/InQuira/install_override.xml
```

To deploy the installation override file:

- place the file in a location from which it can be read by the application server, for example:

```
<Tomcat_home>/common/classes
```

or

```
inquiragw/web-inf/classes
```

CONFIGURING THE SOAP COMMUNICATIONS OPTIONS

You configure SOAP communication by specifying:

- the URL to which you send the SOAP request
- the deployment descriptor

You can specify the URL and deployment descriptor for the SOAP gateway by adding the following configuration to the `installation.xml` file:

```
<root>
  <gateway>
    <soap>
      <config>
        <url>http:// [HOSTNAME] : [PORT] / [WEBAPP PATH] /servlet/rpcrouter</url>
        <deploymentDescriptor>/com/InQira/infra/gateway/soap/
          DeploymentDescriptor.xml</deploymentDescriptor>
      </config>
      <gateway keyref="gateway.gateways [0] " />
    </soap>
  </gateway>
</root>
```

where:

Parameter	Description
<url>	specifies the path to the SOAP servlet within the configured agent in the presentation environment
<deploymentDescriptor>	specifies the location of the deployment descriptor file within the configured agent. The default value is: /com/InQira/infra/gateway/soap /DeploymentDescriptor.xml

SPECIFYING THE URL AND DEPLOYMENT DESCRIPTOR

You can specify the URL and deployment descriptor for the SOAP gateway by adding configuration to the `installation.xml` file:

- add the following to `instance.xml`

```
<root>
  <gateway>
    <soap>
      <config>
        <url>http:// [HOSTNAME] : [PORT] / [WEBAPP PATH] /servlet/rpcrouter</url>
        <deploymentDescriptor>/com/InQira/infra/gateway/soap/
          DeploymentDescriptor.xml</deploymentDescriptor>
      </config>
      <gateway keyref="gateway.gateways [0] " />
    </soap>
  </gateway>
</root>
```

where:

Parameter	Description
<url>	specifies the path to the SOAP servlet within the configured agent in the production web application
<deploymentDescriptor>	specifies the location of the deployment descriptor file. The standard configuration is: /com/InQira/infra/gateway/soap /DeploymentDescriptor.xml

Using the Agents

Oracle Knowledge provides the following integration agents in the Oracle Knowledge archive:

```
<InQuira_home>/archive
```

You use the integration agents by configuring and deploying them as appropriate for your environment.

- The HTML agent, `htmlagent.war`, is a J2SE compliant web application that performs XSLT processing using the provided integrated (Tomcat) web server. This web application is configured as part of the standard configuration, as described in the *Intelligent Search Installation Guide*.
- The J2EE agent, `j2eeagent.war`, is a J2EE compliant web application that performs XSL processing using an Oracle Knowledge gateway. You configure and deploy the J2EE agent, as described in [Using the J2EE Agent on page 83](#).
- The ASP Agent, `aspagent.zip`, provides an IIS/ASP-compliant web application. You configure and deploy the J2EE agent, as described in [Using the ASP Agent on page 85](#).

Using the J2EE Agent

You can use the J2EE agent with the SOAP client.

To use the J2EE agent:

- Configure the appropriate client, as described in [Configuring the SOAP Client for the ASP Agent on page 85](#).
- Deploy the J2EE agent in the production client web server or application server, as described in [Deploying the J2EE Agent on page 84](#).

Configuring the SOAP Client for the J2EE Agent

You configure the SOAP client for use with the J2EE agent and the SOAP gateway by editing the User Interface and Advanced Configuration Facility java server pages to specify the location of the configured SOAP gateway and the deployment descriptor name.

To configure the SOAP client for the J2EE agent:

- 1 Expand the J2EE archive in a temporary directory.
- 2 Edit the User Interface and Advanced Configuration Facility java server pages.
- 3 Recreate the agent archive and deploy the agent.
- 4 Verify the configuration.

Expanding the J2EE Agent Archive

To expand the J2EE agent archive:

- 1 Create a temporary directory, or use any convenient location
- 2 Expand the J2EE agent archive

```
jar -xvf <InQuira_home>/archive/j2eeagent.war
```

The J2EE agent archive expands into the following directory structure:

Directory	Description
css	Contains style sheets for the Java Server Pages (JSP).
images	Contains images for the JSP.
js	Contains javascript files.
META-INF	Contains the manifest file for the archive.
WEB-INF	Contains essential classes and libraries for the agent.
File	Description
admin.jsp	This is the Oracle Knowledge Advanced Configuration Facility.
contact.jsp	This is the JSP for the Contact Deflection module.
ui.jsp	This is the Oracle Knowledge default User Interface.

Specifying the Gateway URL

You configure the SOAP client by editing the Advanced Configuration Facility and User Interface java server pages located in the expanded agent archive, so that they can locate the active SOAP gateway.

To configure the SOAP client

- 1 Open the Advanced Configuration Facility java server page, `admin.jsp` in a text editor
- 2 Locate the following property setting:

```
props.setProperty( com.InQuira.infra.client.soap13.Client.URL, "http://localhost:9999/inquiragw/servlet/rpcrouter" );
```

- 3 Specify the following values:

localhost	Specify the host name of the processor on which the SOAP gateway is installed.
port	Specify the port that the SOAP gateway is configured to listen on.
inquiragw	Specify the name of the gateway that you have installed and configured; for example, the default SOAP gateway is named <code>inquiragw</code> . Note: Ensure that the path to <code>rpcrouter</code> is correct for your installed gateway.

- 4 Specify the same values for this property setting in the file `ui.jsp`.

Deploying the J2EE Agent

To deploy the J2EE agent:

- 1 Re-create the agent archive with the configured Advanced Configuration Facility and User Interface java server pages
- 2 Deploy `j2eeagent.war` into J2SE 1.3-compliant web or application server

Accessing the Advanced Configuration Facility and User Interface

You can access the Advanced Configuration Facility and User Interface from the following URLs:

To access the Advanced Configuration Facility:

- Open the URL:

```
http://host:port/j2eeagent/admin.jsp
```

To access the User Interface:

- Open the URL:

```
http://host:port/j2eeagent/ui.jsp
```

Using the ASP Agent

You can use the ASP agent with the SOAP client.

To use the ASP agent:

- 1 Configure the SOAP client, as described in [Configuring the SOAP Client for the ASP Agent on page 85](#).
- 2 Deploy the ASP agent in the production client web server or application server, as described in

Note: The `InQuira/components` directory contains two components that are required for ASP support, and a README file that provides usage information.

Prerequisites for Using the ASP Agent

Ensure that the following prerequisites are met before using the ASP Agent:

- Microsoft IIS V4 or higher must be installed
- Microsoft XML Core Services V4.0 SP1 or higher must be installed
- Microsoft Windows HTTP Services 5.0 SDK must be installed
- Network access to a properly configured stand-alone Oracle Knowledge 6 SOAP gateway

Note: Please ensure that the required Microsoft tools and components are properly installed, and that you are authorized to use them.

Configuring the SOAP Client for the ASP Agent

To configure the SOAP client for the ASP agent:

- 1 Create a directory to be the location of the web application.
- 2 Expand the ASP agent archive.
- 3 Configure the URL for the SOAP gateway.
- 4 Add and configure a new web application (virtual directory) in the IIS configuration.

For example, to install the ASP agent in:

```
C:\Inetpub\wwwroot\InQuira_web
```

to communicate with the SOAP gateway at:

```
http://soaphost:8888/inquiragw/servlet/rpcervlet
and make the web application accessible at the location:
```

```
http://myhost/InQuira
```

Creating the Application Directory

Create the directory for the web application using the standard procedure for the presentation server environment. For example, in the application directory structure:

```
C:\inetpub\wwwroot
```

- create an application directory Oracle Knowledge_web:

```
mkdir c:\inetpub\wwwroot\InQuira_web
```

Expanding the ASP Agent Archive

To expand the ASP agent archive:

- 1 Copy the packaged ASP agent archive from its location in the Oracle Knowledge archive directory to the application directory, for example:

```
C:\inetpub\wwwroot\InQuira_web\aspagent.jar
```

- 2 Expand the archive in the application directory using an appropriate method for your environment, for example:

```
unzip c:\temp\aspagent.zip
```

The application directory contains the following directories and files:

Directory	Description
css	Contains style sheets for the ASPs.
images	Contains images for the ASPs.
js	Contains javascript files.
xsl	Contains stylesheets for the various components.
File	Description
Admin.asp	This is the Oracle Knowledge Advanced Configuration Facility.
Contact.asp	This is the ASP for the Contact Deflection module.
Default.asp	This is the Oracle Knowledge default User Interface.
error.asp	This is the error page for the User Interface.
global.asa	The <code>global.asa</code> file is an optional file that can contain declarations of objects, variables, and methods that can be accessed by every page in an ASP application

Modifying the SOAP Application Variable

To modify the `soapAddress` application variable:

- 1 Open the file `global.asa` in a text editor.
- 2 Locate the following line:

```
Application( "soapAddress" ) = http://localhost:8080/inquiragw/servlet/
rpcrouter
```

- 3 Specify the value of the variable to be the fully qualified URL of the standalone SOAP gateway, for example:

```
Application( "soapAddress" ) = http://soaphost:8888/inquiragw/servlet/
rpcrouter
```

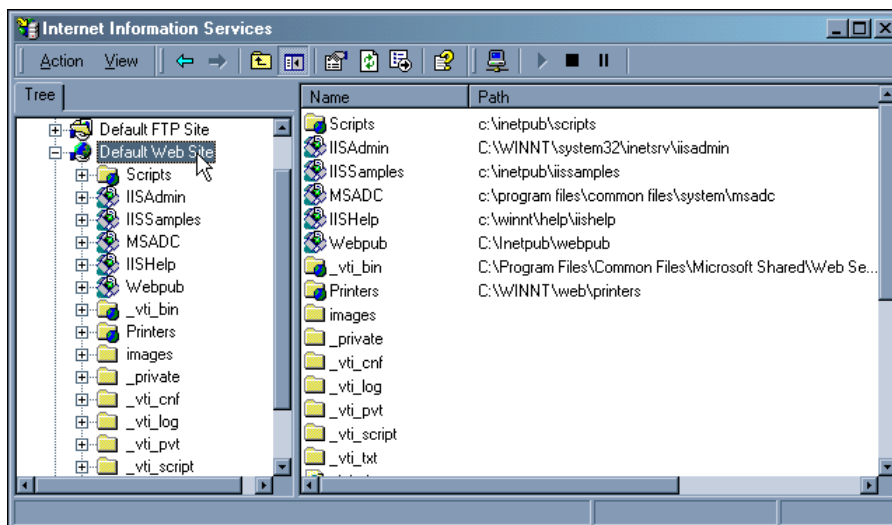
Adding the New Application to the IIS Configuration

Edit the IIS configuration to add a new web application, or virtual directory.

- 1 Start the Windows Management Console for IIS using the appropriate method for your environment; for example:

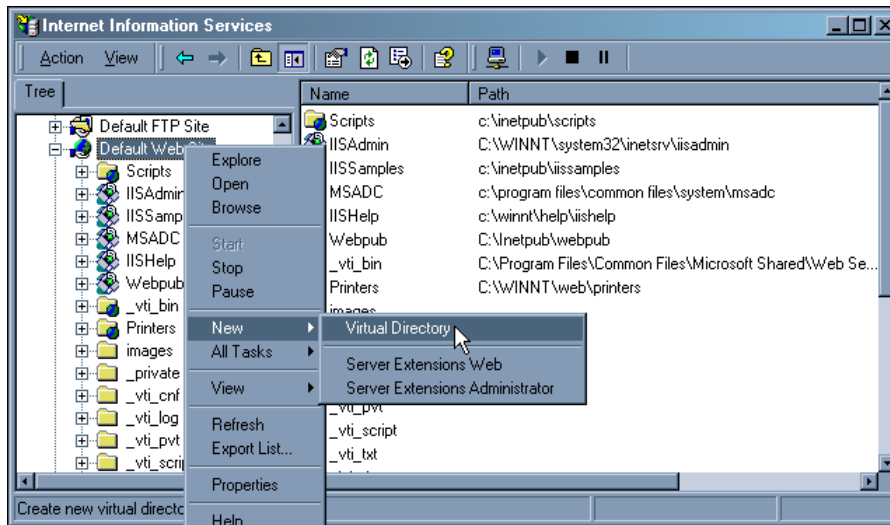
```
%WINDIR%\system32\inetsrv\iis.msc
```

The IIS Management Console window displays the web application hierarchy.



- 2 Right-click to select the appropriate web server (usually Default Web Site)
The IIS menu displays.

3 Select New -> Virtual Directory

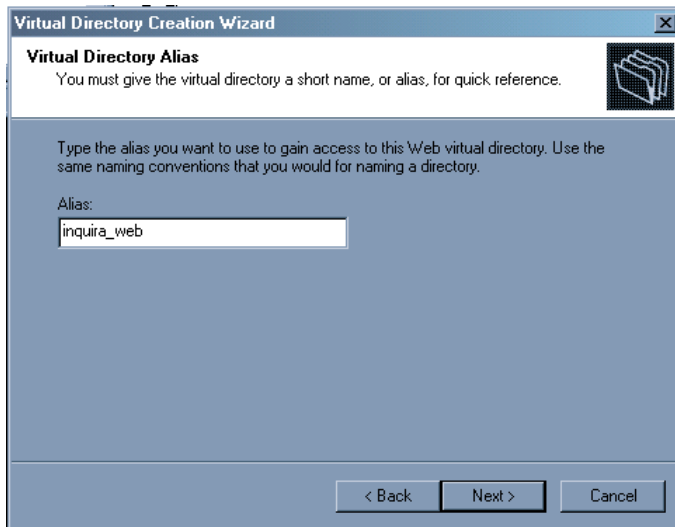


The Virtual Directory Creation Wizard displays:



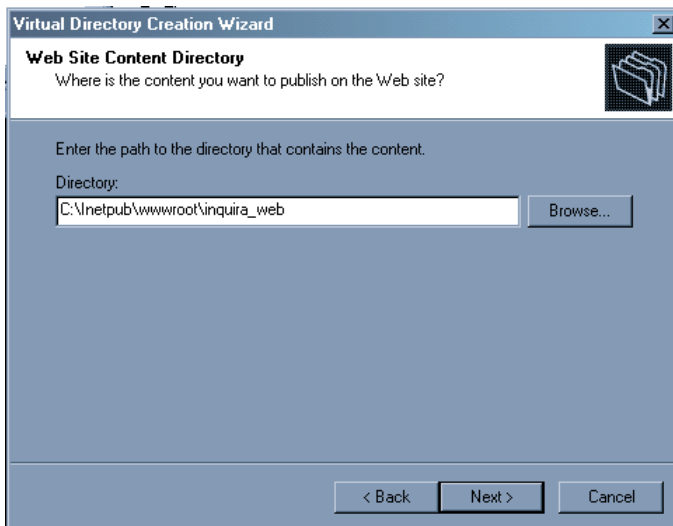
4 Select **Next >** to create a new virtual directory, and enter the Alias:

InQuira_web



- 5 Select **Next >** and enter the Web Site Content Directory:

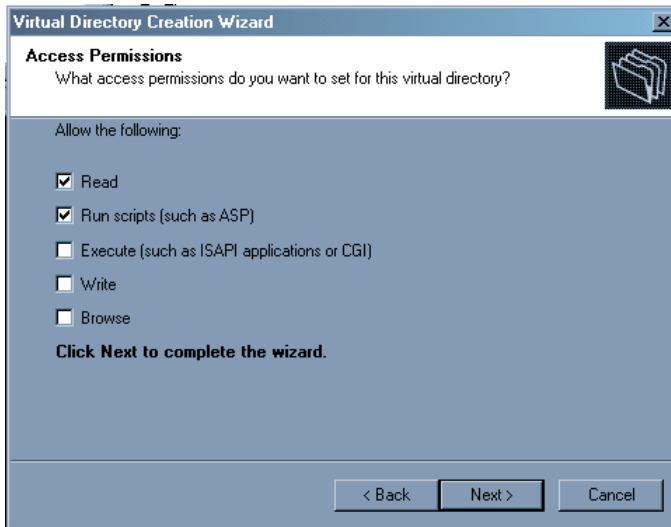
C:\inetpub\wwwroot\InQuira_web



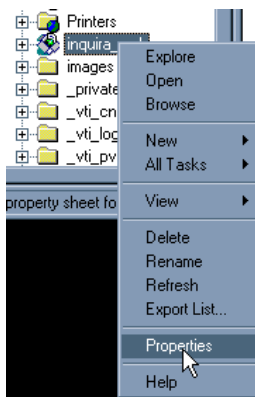
- 6 Enter **Next>** and set the default permissions:

Read

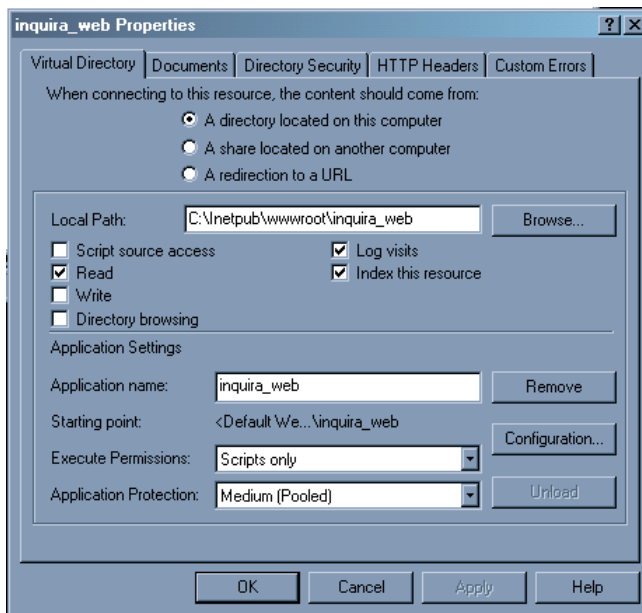
Run scripts (such as ASP)



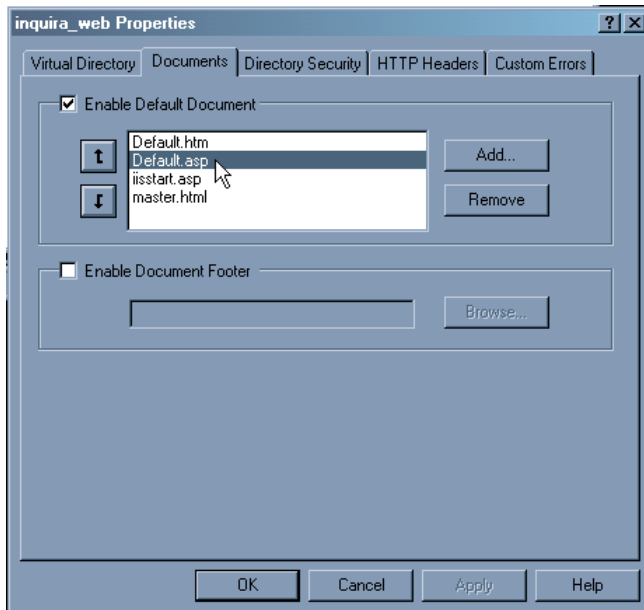
- 7 You can make further modifications by right-clicking on the newly created virtual directory and selecting **Properties**:



The Properties dialog displays:



Note: Ensure that Default.asp is configured as an enabled default document, as shown:



Configuring Access from J2EE Clients

You can access the Oracle Knowledge runtime from a J2SE 1.3 environment using the following methods:

- Using the `com.Inquire.infra.client.Client` API that allows for XML input and output.

- Using the `com.InQira.infra.client.Client` API that allows for XML input that returns fully processed HTML.
- Using J2EE agent XSLT processing on a J2SE 1.3 compliant web or application server.

Configuring Distributed Applications

You can configure an Oracle Knowledge production environment to distribute content processing and request processing work among multiple configured instances, as described in the following sections:

- [Implementing Distributed Content Processing on page 92](#)
- [Implementing Distributed Request Processing on page 95](#)

Implementing Distributed Content Processing

You can configure an Oracle Knowledge production environment to distribute content processing tasks among configured remote Oracle Knowledge instances.

In a distributed content processing environment, a Scheduler instance coordinates the distribution of content processing work to various configured remote work clients. The Scheduler instance can also use its local work client as part of the content processing group.

Important! You cannot distribute content processing work among multiple configured local work clients.

Distributed content processing increases efficiency in a production environment by enabling multiple instances to process a single set of content simultaneously.

The Scheduler instance distributes content processing work by assigning content processing work to remote work clients either on a on a collection basis, or by using optional sub-collections, which help to distribute content processing more equally between multiple work clients.

The Distributed Content Processing Environment

A distributed Oracle Knowledge content processing environment consists of the following configured components:

- A configured scheduler instance that coordinates content processing
- Only one work client residing on the scheduler instance that can perform preprocessing and indexing, and that must perform collection maintenance and global maintenance
- One or more configured remote work clients that perform preprocessing and indexing

Important! You cannot distribute content processing work among multiple configured local work clients.

Distributed Content Processing

In standard (non-distributed) content processing, the scheduler instance communicates directly with the local work client to acquire, standardize, and index the application content.

In distributed content processing:

- The scheduler divides content acquisition work among the configured work clients on a collection basis.
- The remote work clients write the acquisition results to the configured central content store.
- The scheduler assigns preprocessing and indexing work to the configured work clients, either on a collection or sub-collection basis.
- The work clients perform preprocessing and indexing tasks for the assigned content collections or sub-collections, and send the completed work to the scheduler instance.
- The scheduler instance performs the final consolidation (collection maintenance and global maintenance) processes.

Note: You can perform the final index consolidation (index maintenance) only on the scheduler instance.

Content Processing Work Distribution

The Scheduler instance distributes content processing work, as described in the following table:

For this content processing step...	The Scheduler instance...
Content Acquisition (Crawl)	distributes work only by collection
Content Standardization (Preprocess)	distributes work by sub-collection if enabled
Content Indexing	distributes work by sub-collection if enabled
Collection Maintenance and Global Maintenance	uses only its configured local work client

Performing Distributed Content Acquisition (Crawl)

You perform distributed content acquisition by scheduling the standard content acquisition tasks to perform full or incremental acquisition. The scheduler distributes the acquisition task to the configured work clients on a collection basis; the smallest unit of work that can be sent to a work client is a defined content collection.

When the work clients complete the content acquisition process, they send the updated content to the scheduler and notify the scheduler that the work is complete. The scheduler saves the updated collections in the central Content Store.

Performing Distributed Content Standardization (Preprocessing)

You perform distributed standardization (preprocessing) by scheduling either:

- The standard preprocessing tasks to perform full or incremental preprocessing
- The sub-collection preprocessing tasks to perform full or incremental preprocessing using sub-collections

For standard preprocessing, the scheduler distributes the task to the configured work clients on a collection basis; the smallest unit that can be sent to a work client is a single content collection.

For sub-collection preprocessing, the scheduler:

- Distributes the task to the configured work clients on a sub-collection basis

Note: The processes that create sub-collections for preprocessing and indexing are independent of one another. Oracle Knowledge does not maintain sub-collections between content processing steps.

When the work clients complete preprocessing, they send the sub-collections back to the scheduler. The scheduler saves the standardized data to the Content Store by defined content collection.

Performing Distributed Indexing

You perform distributed indexing by scheduling either:

- The standard tasks to perform full or incremental indexing
- The sub-collection tasks to perform full or incremental indexing using sub-collections

For standard indexing, the scheduler distributes the task to the configured work clients on a collection basis; the smallest unit that can be sent to a work client is a single content collection.

For sub-collection indexing, the scheduler:

- Distributes the task to the configured work clients on a sub-collection basis

Note: The processes that create sub-collections for preprocessing and indexing are independent of one another. Oracle Knowledge does not maintain sub-collections between content processing steps.

When the work clients complete indexing, they send the sub-collections back to the scheduler. The scheduler saves the indexed data to the Content Store by defined content collection.

Performing Index Maintenance in a Distributed Content Processing Environment

You perform index maintenance in a distributed content processing environment by scheduling the standard Collection Maintenance and Global Maintenance tasks. Index maintenance is not distributable, and must be performed on the Scheduler instance.

Configuring Distributed Content Processing

You configure distributed content processing by:

- Configuring the local work client to perform the required tasks, as described in [Configuring the Local Work Client for Distributed Content Processing on page 94](#).
- Configuring one or more remote work clients to perform the required tasks, as described in [Configuring Remote Work Clients for Distributed Content Processing on page 95](#).

Configuring the Local Work Client for Distributed Content Processing

To use a local work client for distributed content processing:

- define the local work client, and assign the tasks that it is allowed to perform, as described in [Using Work Clients on page 70](#).

Important! You cannot configure multiple local work clients to perform content preprocessing, indexing, and index maintenance tasks. If multiple local work clients are configured to perform these content processing tasks, the tasks fail.

When configuring distributed content processing, ensure that no local work clients are allowed to perform indexing.

Configuring Remote Work Clients for Distributed Content Processing

To use a remote work client for distributed content processing:

- define the remote work client, and assign the tasks that it is allowed to perform, as described in [Using Work Clients on page 70](#).

The Preprocessing Sub-Collection Creation Process

When you schedule sub-collection preprocessing, the scheduler creates sub-collections based on specified parameters that determine:

- The maximum size of the sub-collection
- The maximum number of documents in the sub-collection

The scheduler uses the created sub-collections to distribute preprocessing work to the work clients. When preprocessing completes, the scheduler saves the pre-processed data to the Content Store, and deletes the sub-collection information.

Note: If you schedule indexing sub-collections, the scheduler uses the process described in [The Indexing Sub-Collection Creation Process on page 95](#).

The Indexing Sub-Collection Creation Process

When you schedule sub-collection preprocessing, the scheduler creates sub-collections based on specified parameters that define:

- The maximum size of the sub-collection as 100Mb.
- The maximum number of documents in the sub-collection as 5,000.

The scheduler uses the created sub-collections to distribute preprocessing work to the work clients. When preprocessing completes, the scheduler saves the pre-processed data to the Content Store, and deletes the sub-collection information.

Note: If you schedule preprocessing sub-collections, the scheduler uses the process described in [The Preprocessing Sub-Collection Creation Process on page 95](#) to re-create sub-collections.

Implementing Distributed Request Processing

You can configure an Oracle Knowledge production environment for distributed request processing, in which one or more separately configured Oracle Knowledge instances perform information retrieval work for the runtime instance(s) within a production web application.

Distributed request processing increases efficiency in a production environment by:

- Increasing the percentage of the indexed application content available in cache memory
- Increasing the percentage of past results maintained in cache memory
- Effectively reducing the size of the content for matching questions to answers

In a distributed request processing environment, production web applications coordinate the distribution of requests to various configured worker instances, and each worker instance performs searches related only to a defined subset of the indexed content.

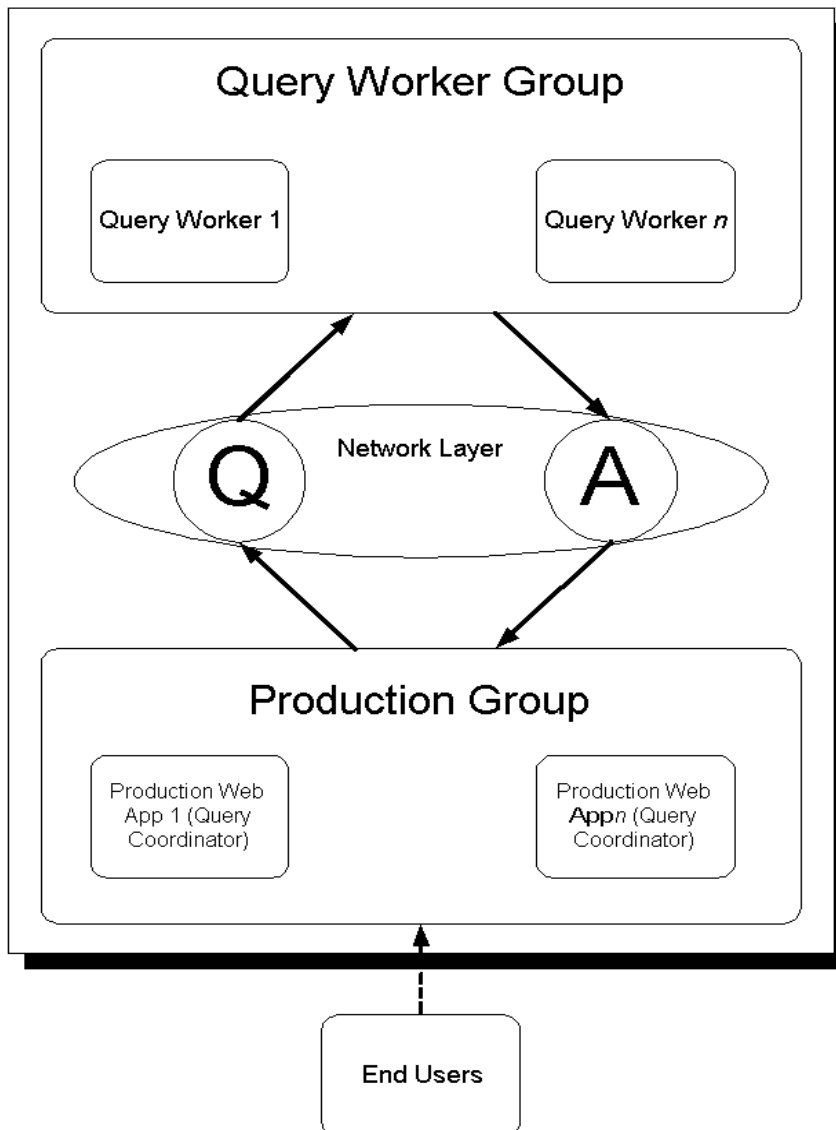
The production web application and the remote worker communicate through a network layer.

Distributed Request Processing Architecture

A distributed Oracle Knowledge runtime environment consists of the following configured components:

- One or more configured production web applications
- A configured query coordinator residing on each production web application, as described in [The Query Coordinator on page 97](#)

- One or more configured query worker instances, as described in [Query Worker Instances on page 98](#)



The Query Coordinator

The query coordinator resides within the production web application and communicates with one or more configured instances called query workers, as described in [Query Worker Instances on page 98](#).

The query coordinator uses the Query Service to:

- package unstructured search requests
- distribute them to remote query workers
- collect and consolidate the query worker results

The Query Service assigns work to the remote query workers on the basis of configured collection groups, as described in [Content Collection Groups on page 98](#).

Query Worker Instances

Query workers are remote instances that you configure to receive work from a query coordinator; you can configure multiple query coordinators and multiple query workers. Query coordinators can send work to multiple workers; however, a query worker can receive work from only one coordinator.

Each configured query worker processes requests for a defined set of content collections, as described in [Content Collection Groups on page 98](#).

Content Collection Groups

Content collection groups are logical groups of defined Oracle Knowledge content collections that you define for use in a distributed Oracle Knowledge request processing environment.

The query coordinator uses content collection groups as the basis for assigning work to the remote query workers. It groups all search requests for a collection group into a single request. Each query worker processes the appropriate requests according to its assigned content collection groups.

Important! A content collection can belong to only one collection group. You cannot assign a content collection to more than one collection group.

You define content collection groups, as described in [Defining Content Collection Groups on page 100](#).

You assign content collection groups to Query Workers, as described in [Assigning Content Collection Groups to Query Workers on page 102](#).

Distributed Request Processing

In a standard (non-distributed) application configuration, the Query Service communicates directly with the local Index Service to satisfy requests for unstructured information retrieval.

In a distributed request processing configuration, the production web application query coordinator communicates with remote query workers through a network layer.

In distributed request processing:

- The query coordinators consolidate queries for each collection group into separate requests, and send them to the network layer
- The query workers process requests and return answers for their assigned content collection groups
- The query coordinator within the production web application collects and presents the answers to the end user

Configuring the Default Distributed Request Processing Implementation

You configure the default distributed request processing implementation for use in a production environment by:

- Configuring a query coordinator in each production web application instance, as described in [Configuring a Query Coordinator on page 99](#).
- Defining one or more content collection groups, as described in [Defining Content Collection Groups on page 100](#).

- Configuring and starting one or more query workers, as described in [Configuring Query Worker Instances on page 101](#).

You configure the distributed request processing components by editing parameters in the configuration file `installation.xml`.

Note: You can include all of the required configuration for distributed request processing in a single `installation.xml` file, which you can then share among all of the query coordinators and query workers in your environment.

Configuring a Query Coordinator

You configure the query coordinator by editing the configuration in the `installation.xml` file to:

- Specify the instance role for the query coordinator instance
- Deploy the query service using the distributed runtime service configuration on a transport with remote capability

To configure a query coordinator, edit the configuration in the `installation.xml` file as shown in the following example:

Example Query Coordinator Configuration

```

<instance name="queryCoordinator" >
  ....
  <role keyref="instanceRole[6]" />
  ....
  <applications index="0" reset="true">
    ....
    <deployedTransport index="0" >
      <transport keyref="instance[queryCoordinator].transports.transport[socket]" />
      <serviceGroup index="0" keyref="serviceGroup[queryServiceOnly]" />
      <serviceConfiguration keyref="serviceConfiguration[distributedQuery]" />
    </deployedTransport>
    <deployedTransport index="1" >
      <transport keyref="instance[queryCoordinator].transports.transport[local]" />
      <serviceGroup index="0" keyref="serviceGroup[runtime]" />
      <serviceConfiguration keyref="serviceConfiguration[default]" />
    </deployedTransport>
  </applications>
</instance>

```

where:

<code>instanceRole [6]</code>	Specifies the query coordinator role
<code>serviceGroup [queryServiceOnly]</code>	Specifies the query service only
<code>serviceConfiguration [distributedQuery]</code>	Specifies the distributed runtime service configuration

Defining Content Collection Groups

Content collection groups are logical groups of content collections that you define for use in a distributed request processing environment. The query coordinator packages search requests based on the collection groups that they belong to, and each query worker processes requests only for its assigned content collection groups.

You can include a single content collection in multiple collection groups to achieve processing redundancy for important content. You can also allocate content collection groups to more than one query worker.

You define collection groups by:

- Specifying the collection group, as described in [Specifying a Collection Group Name](#).
- Specifying the group members, as described in [Specifying Collection Group Members](#).

Specifying a Collection Group Name

You specify a collection group name using the `collectionGroup` parameter:

```
<collectionGroup name="group_name">
```

where:

<code>group_name</code>	specifies the name of the collection group
-------------------------	--

EXAMPLE COLLECTION GROUP NAME CONFIGURATION:

```
<collectionGroup name="nonfiction">
```

Specifying Collection Group Members

You specify the members of the collection group using a `collection index` parameter for each member collection:

```
<collection index="index_position" keyref="content.file-config[collection_name]" />
```

where:

<code>collection index</code>	specifies the position of this group member entry. Valid values are integers, specified such that the initial entry uses the value 0 (zero) and each subsequent entry increments by one integer.
<code>keyref</code>	specifies a defined collection for this member entry. Valid values are any defined content collections.

EXAMPLE COLLECTION GROUP MEMBER CONFIGURATION:

```
<collection index="0" keyref="content.file-config[biography]" />
<collection index="1" keyref="content.file-config[history]" />
```

EXAMPLE COLLECTION GROUP CONFIGURATION

```
<collectionGroup name="nonfiction"> <collection index="0" keyref="content.file-config[biography]" />
<collection index="1" keyref="content.file-config[history]" /> <collection index="2" keyref="content.file-
config[self-help]" /> </collectionGroup>
```

Configuring Query Worker Instances

You configure query worker instances by editing the configuration in the `installation.xml` file to:

- Specifying the instance role for each query worker instance.
- Deploying the query service using the distributed runtime service configuration on a transport with remote capability.

To configure a query worker, edit the configuration in the `installation.xml` file as shown in the following example:

```
<instance name="qw1" >
....
<role keyref="instanceRole[5]" />
....
<applications index="0" reset="true">
....
<deployedTransport index="0" >
  <transport keyref="instance[qw1].transports.transport[socket]" />
  <serviceGroup index="0" keyref="serviceGroup[queryWorkerOnly]" />
  <serviceConfiguration keyref="serviceConfiguration[distributedQuery]" />
</deployedTransport>
<deployedTransport index="1" >
  <transport keyref="instance[qw1].transports.transport[local]" />
  <serviceGroup index="0" keyref="serviceGroup[runtime]" />
  <serviceConfiguration keyref="serviceConfiguration[default]" />
```

```

    </deployedTransport>
    <deployedTransport index="2" >
      <transport keyref="instance[qw1].transports.transport[socket]" />
      <serviceGroup index="0" keyref="serviceGroup[production]" />
      <serviceConfiguration keyref="serviceConfiguration[default]" />
    </deployedTransport>
  </applications>
</instance>

```

where:

<code>instanceRole [5]</code>	Specifies the query worker role
<code>serviceGroup [queryWorkerOnly]</code>	Specifies the query worker only
<code>serviceConfiguration [distributedQuery]</code>	Specifies the distributed runtime service configuration

ASSIGNING CONTENT COLLECTION GROUPS TO QUERY WORKERS

You assign content collection groups to query workers using one of the following methods by specifying either:

- A system property that explicitly names the collection group
- The instance name such that the initial qualifier of the name identifies the collection group

Note: Specifying the system property overrides a specified instance name.

To specify the collection name using the system property, specify:

```
com.InQuira.query.distributed.groupName=<collection_group_name>
```

where:

<code><collection_group_name></code>	Specifies the name of the collection group. If no collection group name is specified, the instance name is used.
--	--

To specify the collection group within the instance name:

- Specify an instance name in the format:
`<collection_group_name>.<instance_ID>`

where:

<code><collection_group_name></code>	Specifies the name of the collection group
<code><instance_ID></code>	Specifies an optional qualifier, which must be delimited by a period (.) for use when specifying a single collection group to be processed by multiple query workers for redundancy and performance optimization.

For example, you could use the following naming convention to configure multiple query worker instances to process the selfhelp collection group:

```

selfhelp.instance1
selfhelp.instance2
selfhelp.instance3

```

STARTING QUERY WORKER INSTANCES

To start a Query Worker, enter the following startup parameters, either manually or as part of a custom startup script:

- 1 Set the InQira_ROOT and CLASSPATH appropriately, using `setenv` or `classpath.bat`
- 2 Set the JAVA_OPTS environment variable
- 3 Use the appropriate standard installation startup script (`run.sh` or `run.bat`), specifying:
 - The instance name
`-Dcom.InQira.infra.instance=<instance_name>`
 - Any appropriate JVM parameters, for example:
`-Xmx1024m`
 - (Optional) the collection group to process
`-Dcom.InQira.query.distributed.groupName=<group_name>`
 - The Instance class:
`com.InQira.infra.Instance -daemon`

For example:

```
-Dcom.InQira.infra.instance="QueryWorker0"
-Xmx1024m
-Dcom.InQira.query.distributed.groupName="selfhelp"
com.InQira.infra.Instance -daemon
```

When distributed runtime system started for the first time following exception occurs:

```
(QUERY_SERVICE_NO_COLLECTION_GROUP queryService)
at com.InQira.query.distributed.transport.QueryServiceImpl.init(QueryServiceImpl.java:92)
at com.InQira.query.distributed.transport.QueryServiceImpl.<init>(QueryServiceImpl.java:68)
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)
at java.lang.reflect.Constructor.newInstance(Unknown Source)
at com.InQira.query.QueryServicesManager.newInstance(QueryServicesManager.java:96)
at com.InQira.revision.AbstractServiceManager.getInstance(AbstractServiceManager.java:129)
at com.InQira.infra.trnsp.TransportBase.locateInternal(TransportBase.java:99)
at com.InQira.infra.trnsp.RemoteTransportBase.locateInternal(RemoteTransportBase.java:80)
at com.InQira.infra.trnsp.socket2.ServiceResponseHandler.processRequest(ServiceResponseHandler.java:305)
at com.InQira.infra.trnsp.socket2.ServiceResponseHandler.run(ServiceResponseHandler.java:118)
at java.lang.Thread.run(Unknown Source)
```

ASSIGNING QUERY COORDINATORS TO QUERY WORKERS

You assign query coordinators to query workers using one of the following methods by specifying either:

- A system property that explicitly names the collection group
- A synchronization group that contains the query coordinator and query worker instances

Note: Specifying the system property overrides a specified instance name.

To specify the collection group using the system property, specify:

```
com.InQira.query.distributed.queryCoordinators
=<query_coordinator_1>[,<queryCoordinator_2>]...
```

where:

<code><query_coordinator_n></code>	Specifies the name of the query coordinator instances that send requests to this query worker. You can specify multiple query coordinators as a comma-separated list.
--	---

To specify query coordinators by synchronization group, define a synchronization group, as described in [Defining the Synchronization Group on page 115](#). The synchronization group must contain all of the query worker instances and all of the query coordinator instances that work with each other.

Note: If you need to isolate query coordinators from one another for distributed request processing, define synchronization groups that contain only the coordinator and worker instances that you want to distribute requests among.

About the Search Component Cache

The cache is built based on the questions asked to the system. The cache is saved to disk (so restarting the instance will not clear the cache). A revision update (such as when ontology or rule changes are made or new content is indexed and synched) updates the cache. The cache files are stored in:

```
<INSTALL DIR>/instances/<INSTANCE NAME>/<ENVIRONMENT ROLE>/<content or runtime>
/applications/default/unstructured/collections/<COLLECTION NAME>/maintenance/
index/revision/<HIGHEST #>/cache/<ENV>/rules_version.dat.
```

Creating Remote Instances for Distributed Applications

You can install and configure remote instances (Work Clients or request processors) that can be controlled from a central scheduler instance. You configure the remote instance by:

- [Installing Oracle Knowledge on the Remote Processor](#)
- [Copying the InQira Common Environment to the Remote Processor](#)
- [Starting InQira Common Environment on the Remote Processor](#)

Installing Oracle Knowledge on the Remote Processor

Install Oracle Knowledge into a directory (having no spaces in the name) on the target processor. We recommend using the same directory structure as the initial installation.

Copying the InQira Common Environment to the Remote Processor

Copy the InQiraCommonEnvironment.jar to the directory:

```
<InQira_install_dir>/packages
```

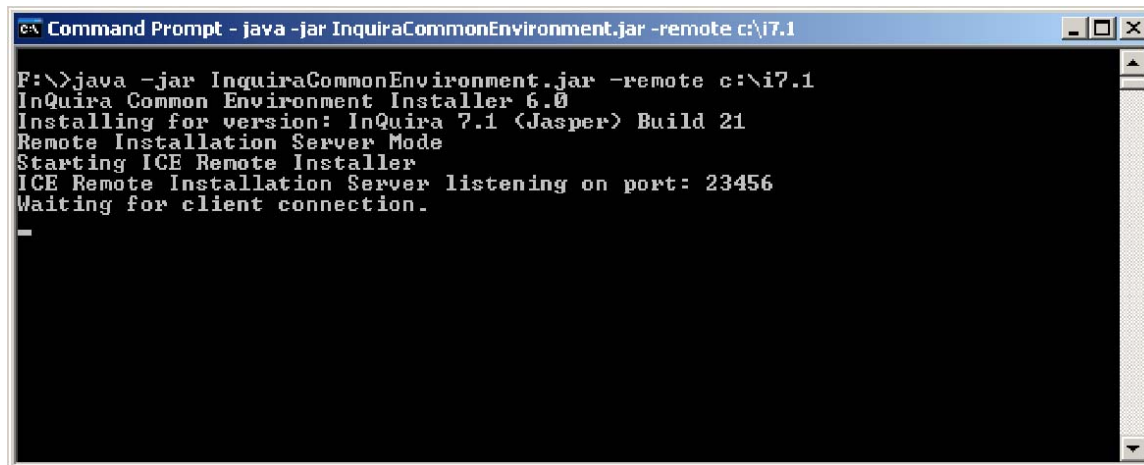
on the host.

Starting Inqira Common Environment on the Remote Processor

Start the Inqira Common Environment installer as a service:

```
java -jar InQiraCommonEnvironment.jar remote <InQira_install_dir>
```

where <InQira_install_dir> is the directory into which you installed Oracle Knowledge.

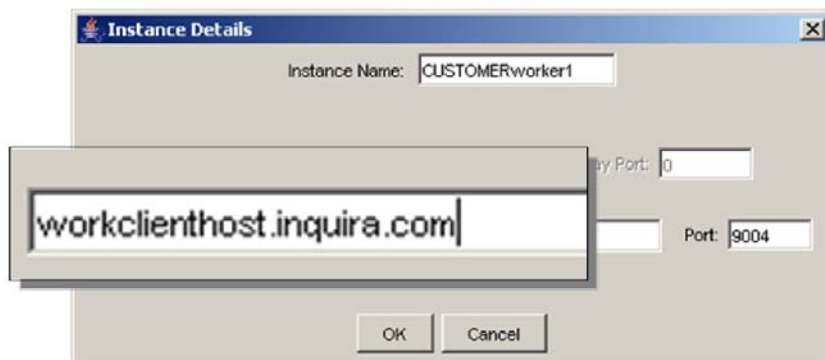


```
Command Prompt - java -jar InQiraCommonEnvironment.jar -remote c:\i7.1
F:\>java -jar InQiraCommonEnvironment.jar -remote c:\i7.1
InQira Common Environment Installer 6.0
Installing for version: InQira 7.1 (Jasper) Build 21
Remote Installation Server Mode
Starting ICE Remote Installer
ICE Remote Installation Server listening on port: 23456
Waiting for client connection.
```

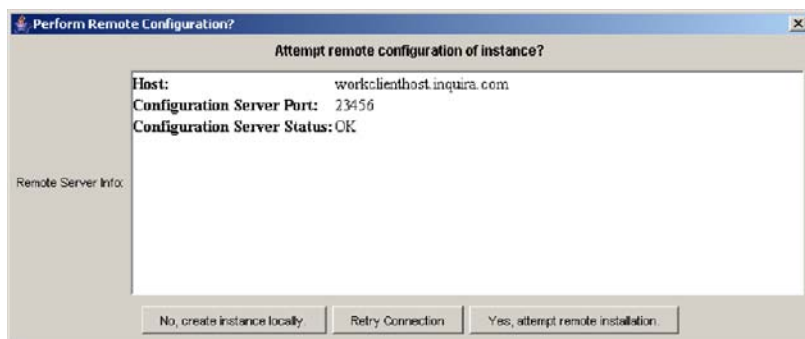
The Inqira Common Environment installer starts as a server and listens for connections on port 23456.

Adding an Instance with a Remote Socket Transport Host

If try to add an instance with a socket transport host that is different from “localhost” then the Inqira Common Environment installer attempts to connect to the remote install instance.



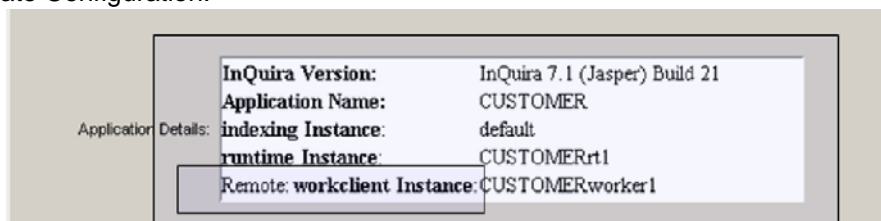
The following dialog displays:



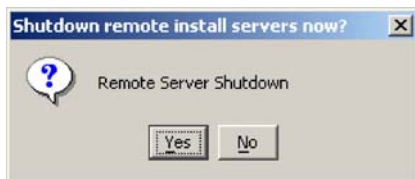
Specifying Remote Installation

If you want the InQuira Common Environment installer to attempt to configure this instance remotely, select: **Yes, attempt remote installation**

The InQuira Common Environment installer lists the remote instance in the Application Details when you select Create Configuration:



When the installation is complete, the ICE installer shows:



Select **Yes** to shut down the remote ICE server process.

Oracle Knowledge Shortcuts in Remote Instances

The InQuira Common Environment installation program installs shortcuts to the setenv scripts appropriate for your environment in the Windows Start Menu:



Moving Data Between Instances

When configuring multiple instances in an Oracle Knowledge production environment, you need to transfer application data, such as indexes, dictionaries, and configuration data, between development, staging, and production runtime instances.

Oracle Knowledge provides the following methods for transferring data between configured instances:

- Synchronization
- Propagation

The synchronization process is designed to transfer the data associated with request and response processing (runtime) from an Oracle Knowledge sending instance to one or more receiving instances; for example, from a scheduler instance to multiple runtime instances.

The propagation process is designed to transfer the data associated with content processing between instances within a processor environment; for example, between development and staging instances.

Revision Levels

The synchronization and propagation processes use internally assigned revision levels to manage the application data during data transfer. Revision levels are integers that increase each time data is saved for a service within the application.

The synchronization process uses revision levels to identify:

- The synchronization data package itself
- The data associated with each service within the package

When you create a synchronization package, the synchronization process maintains revision data for each service, and uses the revision levels to manage the synchronization process.

The synchronization process checks to ensure that the revision level for a given service is lower than the corresponding value in the synchronization package. If a service on a receiving instance has a revision level equal to or greater than the sending instance, the receiving service is not updated.

The Synchronization Process

You use the synchronization process by:

- configuring the various instances within your environment to send and receive data, as described in [Configuring Data Transfer Between Instances on page 108](#).
- performing synchronization as a scheduled task, as described in [Performing Data Transfer on page 120](#).

The synchronization process operates on the sending instance to package the data and assign an incremental value, called a runtime revision level, to the package. The synchronization process also records individual revision levels for each Oracle Knowledge service that supplies synchronization data.

The synchronization process then contacts the receiving instances and switches the receiving instances to maintenance mode, in which they do not accept query requests.

The synchronizer sends the data package to the configured receiving instances over the configured socket, and checks the revision levels for each service. The receiving instance updates its data for any service for which the synchronization package contains newer data, indicated by a higher revision level.

When the required services have been updated, the receiving instance restarts the updated services, notifies the sending instance that synchronization is complete, and returns to runtime mode.

Synchronizing Instances with Minimal Interruption

You can configure additional Oracle Knowledge instances, called controller instances, to enable synchronization with minimal impact on runtime instance availability.

Query Coordinator instances act as receivers during the synchronization data transfer. They write the synchronization data to a file system that is shared with the target runtime instance. When the data transfer process is complete, the controller instance modifies the runtime instance to use the new data from the shared file system.

You configure synchronization with minimal interruption by:

- Installing and configuring one or more controller instances, as described in [Configuring a Query Coordinator Instance for Synchronization with Minimal Interruption on page 114](#).
- Modifying the runtime instances to use the controller instance, as described in [Configuring a Runtime Instance for Use with a Query Coordinator Instance on page 115](#).

The Synchronization Data Package

The synchronizer packages data from the instance from which you issue the synchronize task or command. The data package includes all of the application and configuration data that is required to support request and response processing (runtime) behavior.

Configuring Data Transfer Between Instances

You configure data transfer between instances by setting parameters in the `installation.xml` configuration file, and distributing that file to all of the instances that you want to include in synchronization or propagation.

You specify parameters within the `installation.xml` file to define:

- The sending instance
- The receiving instances
- Optional controller instances to support synchronization with minimal interruption
- Communication for the sending and receiving instances
- The synchronization group

The Installation Configuration File

The installation configuration file contains information about how the various instances within your environment can locate and communicate with one another.

The `installation.xml` file is located in:
`<InQuira_home>/instances/<instance_name>`

Defining Communication for the Sending Instance

The standard mode of communicating between sending and receiving instances is to use a socket connection.

To define a socket connection for the sending instance:

- Edit the `installation.xml` file to define a socket connection, as shown in the following template:


```
<socketTransport name="instance_name">
  <description>instance_description</description>
  <host>host_name</host>
  <port>port</port>
  <secure>>false</secure>
</socketTransport>
```

where:

Parameter	Description
<code>instance_name</code>	Specifies the name of the instance. You can use the default instance, or supply a name for the sending instance.
<code>instance_description</code>	Specifies an optional description of the instance
<code>host</code>	Specifies the name of the processor where the sending instance resides, in the form of a host name or IP address
<code>port</code>	Specifies the port that the sending instance uses. Specify any available port.
<code>secure</code>	Specifies whether to use a secure connection. The default value is false.

Defining Communication for Receiving Instances

The standard mode of communicating between sending and receiving instances is to use a socket connection. To define a socket connection for a receiving instance:

- Edit the `installation.xml` file to define a socket connection, as shown in the following template:


```
<socketTransport name="instance_name">
  <description>instance_description</description>
  <host>host_name</host>
  <port>port</port>
  <secure>>false</secure>
</socketTransport>
```

where:

Parameter	Description
instance_name	Specifies the name of the receiving instance
instance_description	Specifies an optional description of the instance
host	Specifies the name of the processor where the receiving instance resides, in the form of a host name or IP address
port	Specifies the port on which the receiving instance listens for synchronization. Specify any available port.
secure	Specifies whether to use a secure connection. The default value is false.

Example: Data Transfer Transport Definitions

The following is an example of the configuration to define transports for sending and receiving instances:

```
<socketTransport name="sender">
  <description>Sender Machine Socket</description>
  <host>example.1.InQuira.com</host>
  <port>9876</port>
  <secure>>false</secure>
</socketTransport>

<socketTransport name="receiver">
  <description>Receiver Machine Socket</description>
  <host>example.2.InQuira.com</host>
  <port>9765</port>
  <secure>>false</secure>
</socketTransport>
```

Defining the Sending Instance

You define the sending instance within the `installation.xml` file by specifying:

- the instance name, as described in [Specifying the Sending Instance Name on page 110](#).
- the instance mode, which defines the instance as a sending instance, as described in [Specifying the Sending Instance Role on page 112](#).
- the transport to use for communication with receiver instances, as described in [Specifying the Sending Instance Transport on page 112](#).

Specifying the Sending Instance Name

The synchronization process identifies the sending instance by the name that you specify as part of the configuration process.

To specify the name of the sending instance:

- Edit the `installation.xml` file to define the instance name as shown in the following template:


```
<instance name="instance_name">
```

where:

Parameter	Description
instance_name	Specifies the name of the instance

Specifying the Sending Instance Role

The synchronization process identifies sending and receiving instances by an instance mode that you specify as part of the configuration process.

To specify the instance mode for the sending instance:

- Edit the `installation.xml` file to define the instance mode as shown in the following template:
`<role keyref="instanceRole[role_value]" />`

where:

Parameter	Description
<code>role_value</code>	Specifies whether the instance is a sending or receiving instance. Valid values are: <ul style="list-style-type: none"> • 0 - specifies a sending instance • 1 - specifies a receiver instance

Specifying the Sending Instance Transport

You must configure a transport for the sending instance. Specify the transport that you defined in [Defining Communication for the Sending Instance on page 109](#).

To specify the transport for the sending instance:

- Edit the `installation.xml` file to define the transport as shown in the following template:
`<transport name="socket" keyref="transports.socketTransport[transport_name]" />`

where:

Parameter	Description
<code>transport_name</code>	Specifies the name of the sending instance transport

Example: Sending Instance Definition

The following is an example of the configuration to define a sending instance within the `installation.xml` file:

```
<instance name="sender">
  <mode keyref="instanceMode[0]" />

  <transports>
    <transport name="local" keyref="transports.localTransport[internal]" />
    <transport name="socket" keyref="transports.socketTransport[sender]" />
  </transports>

  <synchronizationDataStore>
    <directory>data/synch</directory>
  </synchronizationDataStore>
```


Defining Receiving Instances

You define a receiving instance within the `installation.xml` file by specifying:

- The instance name, as described in [Specifying the Receiving Instance Name on page 113](#).
- The instance mode, which defines the instance as a receiving instance, as described in [Specifying the Receiving Instance Role on page 113](#).
- The transport to use for communication with sending instances, as described in [Specifying Receiving Instance Transports on page 114](#).

You can configure any number of receiving instances for your production environment.

Note: Receiving instances are passive listeners that you configure to receive synchronization packages. Receiving instances do not check for the source of the data that they receive.

Specifying the Receiving Instance Name

The synchronization process identifies receiving instances by the names that you specify as part of the configuration process.

To specify the name of the receiving instance:

- Edit the `installation.xml` file to define the instance name as shown in the following template:
`<instance name="instance_name">`

where:

Parameter	Description
<code>instance_name</code>	Specifies the name of the instance

Specifying the Receiving Instance Role

The synchronization process identifies receiving instances by an instance mode that you specify as part of the configuration process.

To specify the instance mode for a receiving instance:

- Edit the `installation.xml` file to define the instance mode as shown in the following template:
`<role keyref="instanceRole[role_value]" />`

where:

Parameter	Description
<code>role_value</code>	Specifies whether the instance is a sending or a receiver instance. Valid values are: <ul style="list-style-type: none"> • 0 - specifies a sending instance • 1 - specifies a receiving instance

Specifying Receiving Instance Transports

You must configure a transport for the receiving instance. Specify the transport that you defined in [Defining Communication for Receiving Instances on page 109](#).

To specify the transport for the receiving instance:

- Edit the `installation.xml` file to define the transport as shown in the following template:
`<transport name="socket" keyref="transports.socketTransport[transport_name]" />`

where:

Parameter	Description
<code>transport_name</code>	specifies the name of the receiving instances transport that you defined in Defining Communication for Receiving Instances on page 109 .

Example: Receiving Instance Definition

The following is an example of the configuration to define a receiving instance:

```
<instance name="receiver">
  <mode keyref="instanceMode[1]" />

  <transports>
    <transport name="local" keyref="transports.localTransport[internal]" />
    <transport name="socket" keyref="transports.socketTransport[receiver]" />
  </transports>

  <synchronizationDataStore>
    <directory>data/synch</directory>
  </synchronizationDataStore>
```

Configuring a Query Coordinator Instance for Synchronization with Minimal Interruption

You configure a controller instance within the synchronization group as follows:

```
<synchGroup name="nondefault" >
  <redeployAllAtOnce>true</redeployAllAtOnce>
  <synchInstance index="0">
    <instance keyref="instance[sender]" />
    <loadData>false</loadData>
    <saveData>true</saveData>
  </synchInstance>

  <synchInstance index="1">
    <instance keyref="instance[controller]" />
    <loadData>true</loadData>
    <saveData>false</saveData>
  </synchInstance>

  <synchInstance index="2">
```

```

    <instance keyref="instance[receiver]" />
    <loadData>false</loadData>
    <saveData>false</saveData>
  </synchInstance>

```

```
</synchGroup>
```

To specify the instance role for the controller instance:

- Edit the `installation.xml` file to define the instance mode as shown in the following template:


```
<role keyref="instanceRole[4]" />
```

where:

Parameter	Description
4	specifies that the instance is a controller instance

Configuring a Runtime Instance for Use with a Query Coordinator Instance

To configure the runtime instances for use with the controller instance, edit the runtime instance configuration as shown in the following example:

```

<synchInstance index="2">
  <instance keyref="instance[receiver]" />
  <loadData>false</loadData>
  <saveData>false</saveData>
</synchInstance>

```

where:

```
<loadData>false</loadData>
```

specifies that the runtime instance does not receive data during the synchronization process.

Defining the Synchronization Group

A synchronization group defines a sending and receiving configuration for use by the synchronizer. You configure the synch group within the `installation.xml` file.

To configure a synch group:

- Edit the `installation.xml` file to define the synchronization group sending instance as shown in the following template:

```

<synchGroup name="group_name">

  <synchInstance index="0" >

    <instance keyref="instance[instance_name]" />
    <loadData>false</loadData>
    <saveData>true</saveData>
    <service index="0" keyref="bootstrapService[configService]" />
    <service index="1" keyref="service[excerpt]" />
    <service index="2" keyref="service[index]" />
  </synchInstance>

```

```

<service index="3" keyref="service[versionControl]" />
<service index="4" keyref="service[spellcheck]" />
<service index="5" keyref="service[dictionaryStore]" />

```

```
</synchInstance>
```

where:

Parameter	Description
Synch Group name	specifies a name for the synchronization group
Synch Instance index	specifies the index position of this instance within the group. Index positions identify the group members, and must be specified in increments, starting with 0 for the sending instance.
instance keyref	specifies the name of the instance that is associated with this synch instance index
loadData	specifies that this instance sends (load) data during synchronization. Valid values are <code>true</code> and <code>false</code> . Specify <code>true</code> to configure the sending instance.
saveData	specifies that this instance receives (save) data during synchronization. Valid values are <code>true</code> and <code>false</code> . Specify <code>false</code> to configure the sending instance.
service index	specifies the various service data to include in the synchronization data package. Use the default settings for these parameters.

- Define the synchronization group receiving instance as shown in the following template:

```
<synchInstance index="1" >
```

```
  <instance keyref="instance[receiver1]" />
```

```
  <loadData>true</loadData>
```

```
  <saveData>false</saveData>
```

```
  <service index="0" keyref="bootstrapService[configService]" />
```

```
  <service index="1" keyref="service[excerpt]" />
```

```
  <service index="2" keyref="service[index]" />
```

```
  <service index="3" keyref="service[versionControl]" />
```

```
  <service index="4" keyref="service[spellcheck]" />
```

```
  <service index="5" keyref="service[dictionaryStore]" />
```

```
</synchInstance>
```

```
</synchGroup>
```

The following example configuration defines a synch group to synch from the sending instance to a receiver:

```
<synchGroup name="synch1">
```

```
  <synchInstance index="0" >
```

```
    <instance keyref="instance[sender]" />
```

```
    <loadData>false</loadData>
```

```
    <saveData>true</saveData>
```

```

    <service index="0" keyref="bootstrapService[configService]" />
    <service index="1" keyref="service[excerpt]" />
    <service index="2" keyref="service[index]" />
    <service index="3" keyref="service[versionControl]" />
    <service index="4" keyref="service[spellcheck]" />
    <service index="5" keyref="service[dictionaryStore]" />

</synchInstance>

<synchInstance index="1" >

    <instance keyref="instance[receiver]" />

    <loadData>true</loadData>
    <saveData>false</saveData>

    <service index="0" keyref="bootstrapService[configService]" />
    <service index="1" keyref="service[excerpt]" />
    <service index="2" keyref="service[index]" />
    <service index="3" keyref="service[versionControl]" />
    <service index="4" keyref="service[spellcheck]" />
    <service index="5" keyref="service[dictionaryStore]" />

</synchInstance>

</synchGroup>

```

Distributing the Installation Configuration File

To distribute the configured `installation.xml` file to the required locations:

- Copy the file `installation.xml` from the sending instance to the web server directory `<tomcat_home>/common/classes` on all configured (sending and receiving) instances
- Add the `installation.xml` file to the classpath on each receiving instance, for example:
`export CLASSPATH=<tomcat_home>/common/classes:$CLASSPATH`

Verifying the Data Transfer Configuration

To validate each configured instance, start the Tomcat web servers and verify the instance definitions in the web server log output:

- Start the Tomcat web server
- Open the web server log output file (`catalina.out`)
- Locate the log entries to verify the correct instance name

The Propagation Process

You use the propagation process by:

- Configuring the various instances within your environment to send and receive data, as described in [Configuring Data Transfer Between Instances on page 108](#).
- Performing propagation as a scheduled tasks, as described in [Performing Data Transfer on page 120](#).

Propagation Process runs on a content processing instance of a source environment to transfer data to a content processing instance on the destination environment.

The receiving instances update data for all applicable services, and the revision levels for all services on the receiving instances are reset to 0.

The Propagation Data Package

The propagation process packages data from the instance from which you issue the propagate task or command. Propagation Data package only contains dictionary data (vcroot) and configuration data (#.xml).

Configuring Propagation

You can configure the propagation process to move data between Oracle Knowledge environments.

Configuring propagation requires that you install and configure Oracle Knowledge and Inquira Common Environment in each environment (typically development and production) as a complete Oracle Knowledge Enterprise Application environment.

Important! We do not recommend that you can create all environments on the same processor; however, if you do, ensure that you correctly define each application server and socket port so that there are no conflicts. Note that this requires different configurations for each environment, which compromises the purpose of maintaining discrete environments.

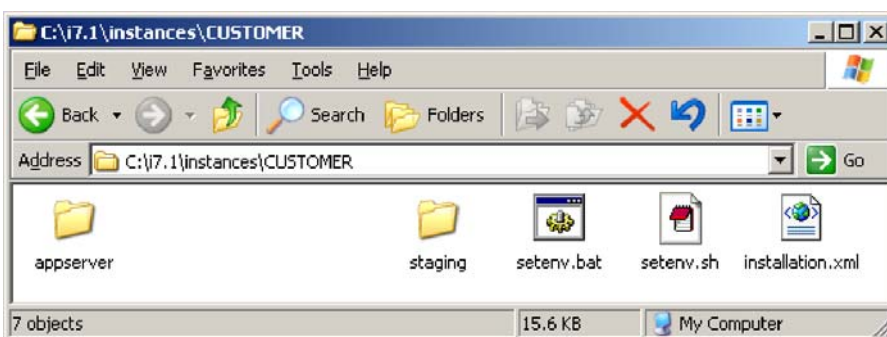
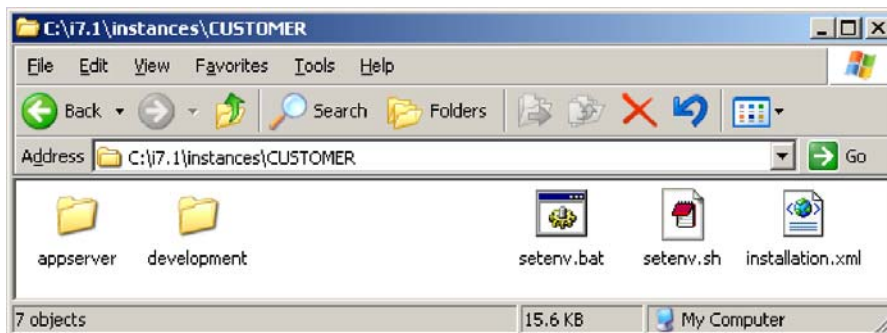
You may not, however, locate all environments in the same directory. To host more than one environment on a single processor, you must create different installation roots (\$InQuira_base_dir) for each environment.

Performing Propagation

To configure propagation, perform the Inquira Common Environment installation and select the appropriate Environment Role:



This produces a configuration and a directory structure identical to the other environment roles installation, except configuration files are written to `./development`, `./staging`, or `./staging` directories:



To enable propagation:

- Create the appropriate `./development`, `./staging`, or `./production` directories in each of the configuration instance directories
- Copy the `env.xml` files from the target environment's directories

For example, to configure propagation from:

- A development environment on:
Hosts `DEVCONTENT1`, `DEVRUNTIME1`

to:

- A production environment on:

PRODCONTENT1 PRODRUNTIME1

- Create the application using the Inqira Common Environment program for Development on DEVCONTENT1 and DEVRUNTIME1
- Create the application using the Inqira Common Environment program for Production on PRODCONTEN1 and PRODRUNTIME1
- Create /<BASE_DIR>/instances/<INSTANCE>/production on DEVCONTENT1
- Create /<BASE_DIR>/instances/<INSTANCE>/production on DEVRUNTIME1
- Copy /<BASE_DIR>/instances/<INSTANCE>/production/env.xml files from PRODCONTENT1 and PRODRUNTIME1 into the /<BASE_DIR>/instances/<INSTANCE>/production directories

You can now schedule the Propagate to Production or Propagate to Staging tasks.

Performing Data Transfer

You can perform synchronization and propagation using the scheduler, as described in [Chapter 4, Operating Oracle Knowledge and Scheduling Jobs](#) to schedule the appropriate synchronization and propagation tasks, as described in [Environment Communication Tasks on page 62](#).

Note: Ensure that you have met the prerequisites for initial synchronization, as described in [Prerequisites for Initial Synchronization on page 120](#).

Prerequisites for Initial Synchronization

Before synchronizing data between sending and receiving instances for the first time, ensure that:

- The application/data directory in each receiver instance is empty
- You have successfully performed content processing on the sending instance

Note: It is also recommended that you test question-answering on the sending instance.

Implementing Oracle Knowledge User Authorization

You can use one or both types of user authorization within your Oracle Knowledge environment:

- The default internal user authorization
- Integration with an external LDAP system

The default internal user authorization system provides the means to:

- Create and manage user IDs and passwords
- Associate user IDs with various permissions

Oracle Knowledge supports integration with external LDAP systems by providing:

- Pre-defined authorization Roles, as described in [Using Oracle Knowledge Authorization Roles on page 125](#).
- Pre-defined access and operational Permissions, as described in [Pre-Defined Permissions on page 125](#).

You integrate Oracle Knowledge user authorization with an LDAP system by:

- Defining an LDAP authentication domain within the Oracle Knowledge environment, as described in [Defining an LDAP Authentication Domain on page 121](#).
- Optionally creating and modifying Oracle Knowledge Roles, as described in [Creating and Modifying Authorization Roles on page 127](#).
- Assigning Oracle Knowledge Roles to groups within your LDAP system

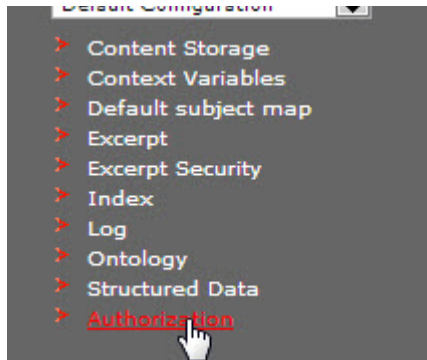
You can use the default internal user authentication in conjunction with one or more configured LDAP Domains. Oracle Knowledge login dialogs display a Domain field from which users can select the internal domain, or an LDAP domain as appropriate.

Defining an LDAP Authentication Domain

You integrate with an external LDAP system by defining an LDAP domain within the Oracle Knowledge environment. You define an LDAP domain within Oracle Knowledge using the Authorization page of the Advanced Configuration Facility.

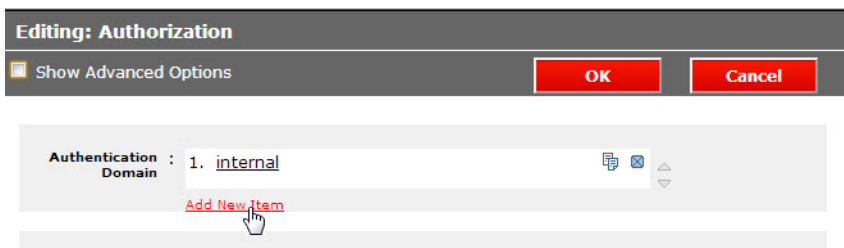
To define an LDAP domain:

- 1 Select **Authorization** from the Instances section of the Advanced Configuration Facility main menu:



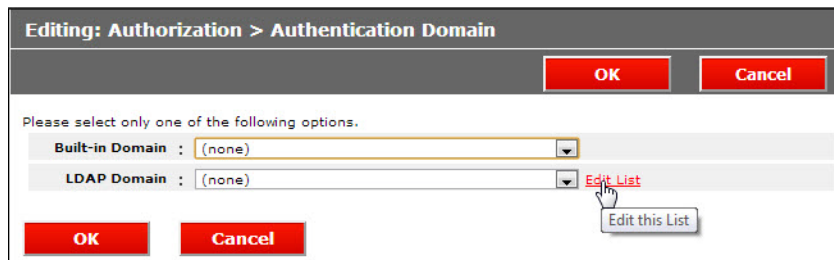
The Authorization page displays.

- 2 Select **Edit**, then select **Add New Item** for the Authentication Domain field:



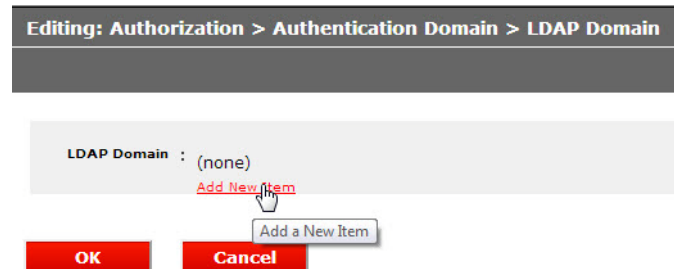
The Authentication Domain page displays.

- 3 Select the **Edit List** item for the LDAP Domain field:



The LDAP Domain page displays.

- 4 Select **Add New Item** for the LDAP Domain field:



The LDAP Domain page displays.

- 5 Specify the LDAP domain properties, as described in [Specifying LDAP Authentication Domain Properties](#) on page 123.

Specifying LDAP Authentication Domain Properties

You define an LDAP authentication domain within the Oracle Knowledge application using the Advanced Configuration Facility LDAP Domain page:

Editing: Authorization > Authentication Domain > LDAP Domain > LDAP Domain

Show Advanced Options OK Cancel

Domain

Provider Class

URL

User Base DN

OK Cancel

- Specify the following parameters:

Parameter	Description
Domain	Specify a name for the LDAP domain
Provider Class	Specify an alternate class to provide LDAP information if required. The default class is: <code>com.sun.jndi.ldap.LdapCtxFactory</code>
URL	Specify the location of the target LDAP system
User Base DN	Specify alternate or additional LDAP hierarchy mapping if required. The defaults are: <ul style="list-style-type: none"> • CN=Users • DC=company • DC=net

Specifying Advanced LDAP Authentication Domain Parameters

You can specify the following optional advanced LDAP authentication domain parameters, as required for your integration, using the Advanced Configuration Facility LDAP Domain page:

Advanced Options

Class Name :	<input type="text"/>
Authentication Method ▶	<input type="text" value="simple"/>
use SSL ▶	<input checked="" type="radio"/> On <input type="radio"/> Off
User Lookup Attribute ▶	<input type="text" value="sAMAccountName"/>
Fully Qualified User DN Attribute ▶	<input type="text" value="distinguishedName"/>
User Display Name Attribute ▶	<input type="text" value="displayName"/>
User Group Attribute ▶	<input type="text" value="memberOf"/>
Reload Interval :	<input type="text" value="-1"/>
Role Mapping DN :	<input type="text"/>
Role Name Attribute :	<input type="text" value="name"/>
Role Mapping Attribute :	<input type="text" value="member"/>

Authentication Properties

Value : (none)
[Add New Item](#)

initialAuthInfo

initialAuthType :

initialAuthPrincipal :

initialAuthCredentials :

initialAuthProps

Value : (none)
[Add New Item](#)

field2ContextNameMap

Value : (none)
[Add New Item](#)

Using Oracle Knowledge Authorization Roles

Oracle Knowledge is installed and configured with pre-defined Roles that you can use to integrate with an external LDAP system.

Note: The authorization roles are not used within the default internal user authorization system.

Each Role is pre-configured with a set of permissions that define its capabilities. You can create and modify authorization Roles, as described in [Creating and Modifying Authorization Roles on page 127](#).

The following Roles are installed and configured as part of the standard installation process:

- Administrator, as described in [Using the Administrator Role on page 126](#)
- Language Administrator, as described in [Using the Language Administrator Role on page 126](#)
- Language Development, as described in [Using the Language Development Role on page 126](#)

Pre-Defined Permissions

Oracle Knowledge is installed and configured with the following Permissions:

Permission	Description
Perform Rules Maintenance	Grants the ability to update Rules within the Oracle Knowledge Language Workbench Dictionary Manager application.
Perform User Maintenance	Grants the ability to update User Manager accounts within the Oracle Knowledge Language Workbench.
Run TestDrive tool	Grants the ability to test request-response processing using the Oracle Knowledge Language Workbench Test Drive application.
Run regression tool	Grants the ability to test request-response processing using the Oracle Knowledge Language Workbench Quality Monitor application.
Maintain top level domains	Grants the ability to create and modify base- and industry-level domains using the Dictionary Manager application.
Maintain domain groups	Grants the ability to create and modify domain groups using the Dictionary Manager application.
Maintain domains	Grants the ability to create and modify customer-level domains using the Dictionary Manager application.
Run ontology builder tool	Grants the ability to operate the Dictionary Manager application within the Oracle Knowledge Language Workbench.
Run dynamic navigation configuration tool	Grants the ability to operate the Personalized Navigation application within the Oracle Knowledge Language Workbench.
Run system administration console	Grants the ability to operate the Advanced Configuration Facility.

Using the Administrator Role

The Administrator role contains the following access and operational permissions:

- Perform rules maintenance
- Perform user maintenance
- Run Test Drive tool
- Run regression tool
- Maintain top level domains
- Maintain domains
- Run ontology builder tool
- Run dynamic navigation configuration tool
- Run system administration tool

See [Creating and Modifying Authorization Roles on page 127](#) for information on adding or removing permissions for this Role.

Using the Language Administrator Role

The LanguageAdministrator role contains the following access and operational permissions:

- Maintain top level domains
- Maintain domain groups
- Maintain domains
- Run analytics administration tool
- Run ontology builder tool
- Run dynamic navigation configuration tool

See [Creating and Modifying Authorization Roles on page 127](#) for information on adding or removing permissions for this Role.

Using the Language Development Role

The Administrator role contains the following access and operational permissions:

- Perform rules maintenance
- Run TestDrive tool
- Run regression tool

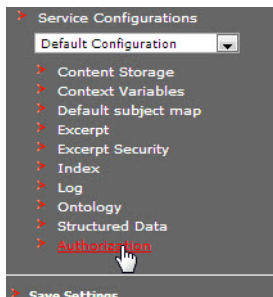
See [Creating and Modifying Authorization Roles on page 127](#) for information on adding or removing permissions for this Role.

Creating and Modifying Authorization Roles

You can create new, and modify existing, authorization roles for use with an integrated LDAP system. You create and modify authorization roles by adding or deleting the permissions that define its capability using the Roles page of the Advanced Configuration Facility.

To add or modify a Role:

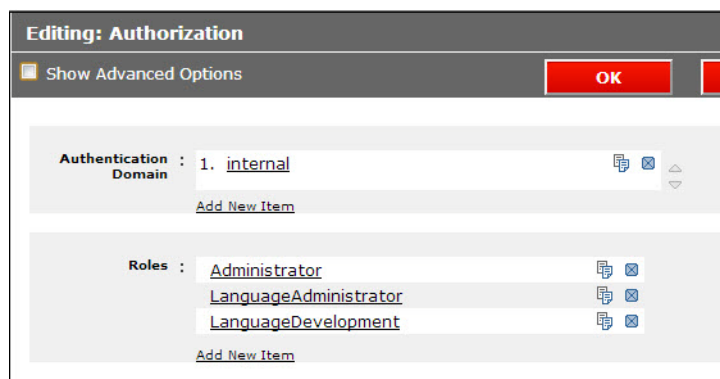
- 1 Select **Authorization** from the Instances section of the Advanced Configuration Facility main menu:



The Authorization page displays.

- 2 Select **Edit**, then select the desired Role, or select **Add New Item** for the Authentication Domain field.

The Roles page displays:



To define a new Role:

- 1 Specify a name for the Role
- 2 Use the **Add New Item** function to add Permissions to the new Role

To modify an existing Role:

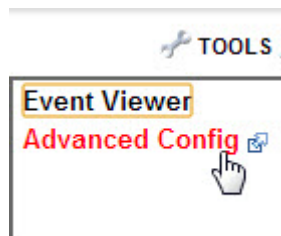
- 1 Use the **Add New Item** and Delete [X] function to modify the list of Permissions for the Role

Advanced Configuration Facility

System Manager contains the Advanced Configuration Facility (Advanced Config). The Advanced Configuration Facility provides a means to modify the required or optional Oracle Knowledge settings.

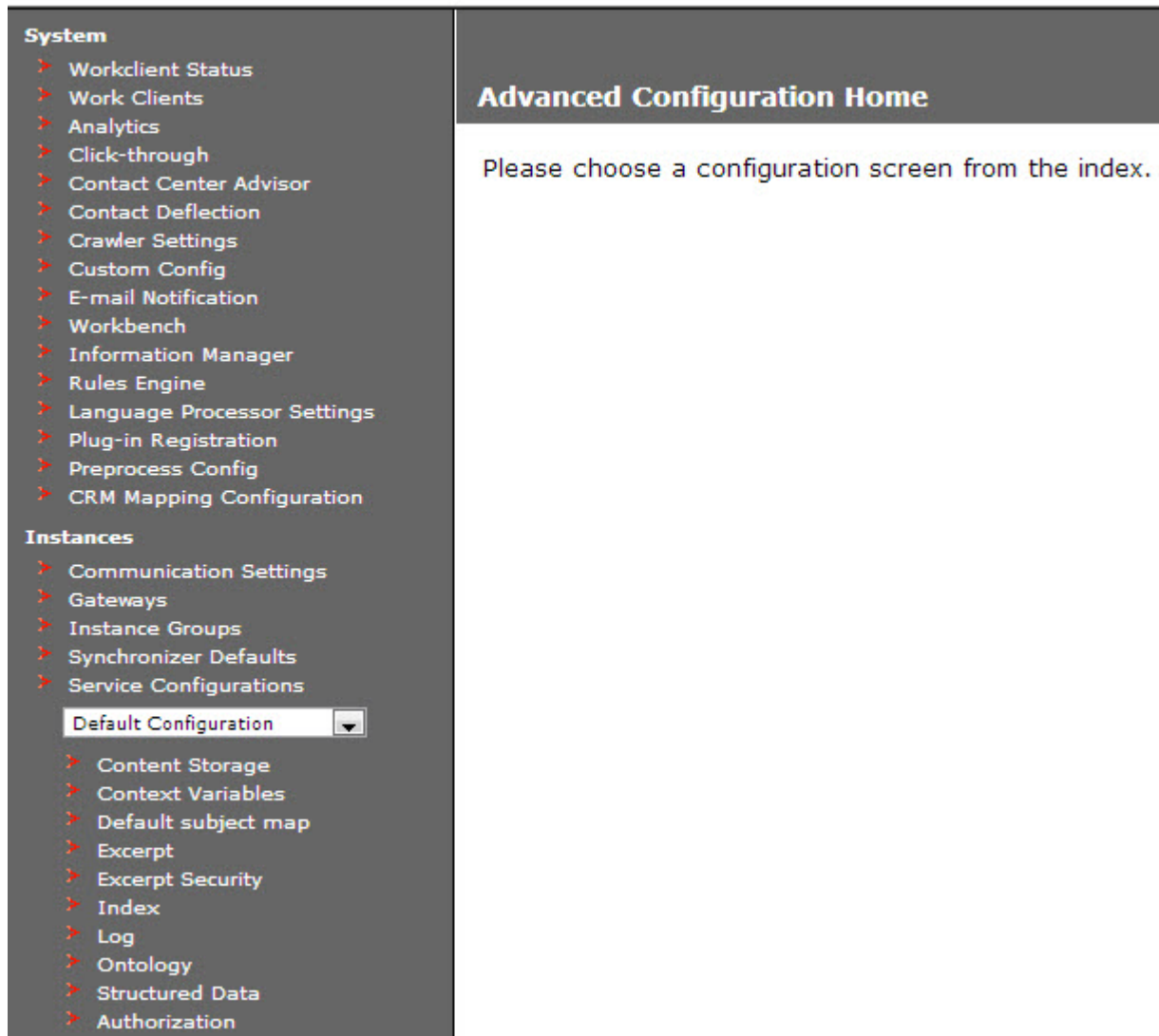
To access Advanced Configuration Facility:

- 1 From System Manager select **Tools**.
- 2 Select **Advanced Config**.



The Advanced Configuration screen displays.

ORACLE[®] Advanced Configuration



System

- Workclient Status
- Work Clients
- Analytics
- Click-through
- Contact Center Advisor
- Contact Deflection
- Crawler Settings
- Custom Config
- E-mail Notification
- Workbench
- Information Manager
- Rules Engine
- Language Processor Settings
- Plug-in Registration
- Preprocess Config
- CRM Mapping Configuration

Instances

- Communication Settings
- Gateways
- Instance Groups
- Synchronizer Defaults
- Service Configurations

Default Configuration ▾

- Content Storage
- Context Variables
- Default subject map
- Excerpt
- Excerpt Security
- Index
- Log
- Ontology
- Structured Data
- Authorization

Advanced Configuration Home

Please choose a configuration screen from the index.

Advanced Config is divided into two parts:

- [System](#) on page 130.
- [Instances](#) on page 147.

System

The System portion of the Advanced Configuration Facility provides a means to modify settings that affect Oracle Knowledge as a whole. For more information see the following sections:

[Work Client Status on page 131](#)

[Work Client on page 131](#)

[Click-Through on page 132](#)

[Contact Center Advisor on page 134](#)

[Contact Deflection on page 136](#)

[Crawler Settings on page 137](#)

[Custom Configuration on page 139](#)

[E-mail Notification on page 139](#)

[Workbench on page 140](#)

[Information Manager on page 140](#)

[Rules Engine on page 141](#)

[Language Processor Settings on page 143](#)

[Plug-in Registration on page 144](#)

[Preprocess Configuration on page 144](#)

Work Client Status

The Work Client Status section of Advanced Configuration Facility allows you to start and stop configured Oracle Knowledge Instances. This is particularly useful for accessing remote instances.

After selecting Work Client Status from Advanced Configuration Facility the Work Client Status screen displays.

On/Off	Type	Name	Status
<input checked="" type="radio"/> On <input type="radio"/> Off	Local	Local Workclient	Running
	Remote	MLD24worker1	Stopped

To stop or start an instance, select the on/off option for a particular instance.

Note: It may take up to 2 minutes to start or stop a work client.

Work Client

You can add a work client to any Oracle Knowledge instance. You can define work clients as local or remote work clients. Remote work clients are configured on an instance that is remote from the central scheduler instance. Once you have configured a remote work client, you can start and stop it from the remote scheduler instance.

You add a work client by:

- 1 Defining a work client
- 2 Configuring allowed tasks for the work client
- 3 (For remote work clients) defining the work client for a remote scheduler

To define a work client:

- 1 Select **Work Client** from the Advanced Configuration Facility menu.
- 2 Select **Edit**.
The Work Clients page displays information about the currently defined work clients, and an Add New Item link.
- 3 Select the **Add New Item** link.

The Work Clients page displays:

4 Enter the following information:

Parameter	Description
Item Name	Enter a name for the work client. This is the name that displays in the work client status screen.
Instances	Select Add New Item Enter the name for the Instances specified in the Item Name field.
Allowed Tasks	Select the Add New Item link to add tasks that this work client is authorized to perform.

Click-Through

Selecting Click-through from Advanced Configuration Facility allows you to set and modify the settings for click-through tracking.

By selecting **Click-Through, Show Advanced Options, Edit** the Editing: Click-through screen displays.

Editing: Click-through

Show Advanced Options OK Cancel

Perform HTML Highlighting : On Off

performPDFHighlighting : On Off

Perform Click-through Tracking : On Off

Perform Search Within Document : On Off

Perform Default Question Highlight : On Off

HTML Highlighting

Highlight Title Style : color:#000000; background:#E8F5FF

Highlight Sentence Style : color:#000000; background:#E8F5FF

Honor Document Anchor : On Off

Check For Location Replace : On Off

Allow Automatic HTTP Redirects : On Off

Cookie Format Policy : Legacy Format [Edit List](#)

timeoutSetting : 10000

contentSource : 1. [IM](#)

Perform HTML Highlighting

Specifies whether to highlight answer text within linked answer documents. The default value is on.

Perform PDF Highlighting

Specifies whether to highlight answer text within linked answer documents. The default value is on.

Perform Click-through Tracking

Specifies whether to record end-user selection of answer links for use by Analytics. The default value is on.

Perform Search Within Document

Specifies whether to perform search within documents. The default value is off.

Perform Default Question Highlight

Specifies whether to perform question highlighting. The default value is off.

Highlight Title Style	Specifies the HTML style parameters for title highlighting. The default values specify a font color #000000 (black), and a background color #E8F5FF (bright cyan)
Highlight Sentence Style	Specifies the HTML style parameters for sentence highlighting. The default values specify a font color #000000 (black), and a background color #E8F5FF (bright cyan)
Honor Document Archive	Specifies whether to use the predetermined closed anchor in an HTML document. If this feature is turned off, the browser jumps to an artificial anchor used in highlighting. The default value is off.
Check For Location Replace	Specifies whether to check the document for location.replace java script. If this feature is turned on, the document does not contain highlighting if this script is found. The default value is off.
Allow Automatic HTTP Redirects	Specifies whether to allow automatic HTTP redirects. The default value is off.
Cookie Format Policy	Specifies cookie policy when highlighter (httpclient) is trying to acquire linked answer documents for highlighting. The default is Legacy Value. To modify select Edit List .
timeoutSetting	Specifies the timeout limit when highlighter (httpclient) is trying to acquire linked answer documents for highlighting. The default is none. To modify select Edit List .
contentSource	Allows you to specify which collection types should be highlighted, regardless of the document being in HTML. To modify select Add New Item .
Username	Refers to the Username associated with the Proxy Server used for highlighter (httpclient)
Password	Refers to the password associated with the Proxy Server used for highlighter (httpclient)
Domain	Refers to the Domain used by the associated Proxy Server.
Proxy Host	Refers to the Host URL used by the associated Proxy Server.
Proxy Port	Refers to the port used by the associated Proxy Server.

Contact Center Advisor

Selecting Contact Center Advisor from Advanced Configuration Facility allows you to set and modify the settings for the Oracle Knowledge Contact Center Advisor.

By selecting **Contact Center Advisor, Show Advanced Options, Edit** the Editing: Call Center Advisor screen displays.

cca-default-handler	Refers to how Oracle Knowledge communicates with CCA installations. See Call Center Advisor below. To modify select Edit List .
Call Center Advisor	Refers to the CRM system used with CCA. The default is Siebel. See cca-handler-impl below.
Base URL	Refers to the URL associated with the configured CRM system.
User Name	Refers to the login name associated with the CRM server.
Password	Refers to the password used to access the CRM server.
Call Center Advisor	Refers to how Oracle Knowledge communicates with CCA. To modify select Add New Item .
cca-handler-impl	Defines the code used by Oracle Knowledge to invoke communication with the CRM. To modify select Add New Item .

Contact Deflection

Selecting Contact Deflection from Advanced Configuration Facility allows you to set and modify the settings for contact deflection. By selecting **Contact Deflection, Show Advanced Options, Edit** the Editing: Contact Deflection screen displays.

Item Name	Lists the names of the available contact deflection methods. The default is email.
Domain Group	Refers to the dictionary domain group active for e-mail deflection. The default is Contact, which contains the packaged contact deflection rules
Handler Class	Code to recall Contact Page.
Value	To modify select Add New Item .
Escalation URL	Specifies the URL associated with the Escalation process.

Escalate with no answers	Specifies whether to automatically redirect to the Escalation page specified above, when no search results are returned. The default value is on.
Handler Class	Code used to recall Escalation Page
Value	To modify select Add New Item .

Crawler Settings

Selecting Crawler Settings from Advanced Configuration Facility allows you to see and modify the settings for the crawlers used in Oracle Knowledge. By selecting **Crawler Settings, Show Advanced Options, Edit** the Editing: Crawler Settings screen displays.

Editing: Crawler Settings

Show Advanced Options OK Cancel

Date Format :

Database Crawlers : (none)
[Add New Item](#)

Siebel Crawlers : (none)
[Add New Item](#)

Custom Crawlers : (none)
[Add New Item](#)

Document Attribute : (none)
[Add New Item](#)

Collection Group : (none)
[Add New Item](#)

Content Acquisition

Commit Interval :

Date Format	Specifies any valid date formats that the application must process in addition to the standard packaged date formats. The standard date formats are:<day-abbr>, <day-of-month> <name-of-month> <year> <24hour>:<minutes>:<seconds> <timezone>
Database Crawler	Allows you to crawl configured databases. To modify select Add New Item
Siebel Crawlers	Allows you to crawl configured Siebel systems. To modify select Add New Item .
Custom Crawlers	Allows you to crawl configured custom content. To modify select Add New Item .
Document Attribute	Lists predefined document attributes to be crawled. To modify select Add New Item .
Collection Group	Lists predefined collection groups to be crawled. To modify select Add New Item
Commit Interval	Saves index documents after every X number. The default value is 500.
Download URL Patterns	Allows you restrict the type of documents to be crawled. To modify select Add New Item .
Document Attribute Selector	Allows you to set custom code for Document Attributes to be crawled as defined above.
Time Frame	Allows you to configure crawls based on a date

Tip: Reducing the Index Data and Speeding Up Indexing

You can reduce the index data and speed up index processing using a configurable limit for large documents.

To put a limit on the index data size for large documents, insert the following code, in red, into the .xml configuration file:

```
<index>
  <debug>false</debug>
  <track-unknowns>true</track-unknowns>
  <max-sentence-length>75</max-sentence-length>
  <!-- This is index cut-off in Byte per document -->
  <max-index-doc-length>4194304</max-index-doc-length>
</index>
```

Note:

- The installed default is 4MB cut-off in bytes from IQXML data per document. The size limit is not on the raw document, not on the final indexed data - but an intermediate file IQXML.
- This configuration applies to all the documents, regardless of document type or collection.
- Value of "-1" means no cut-off in byte. The entire IQXML is indexed.

Custom Configuration

Selecting Custom Configuration from Advanced Configuration Facility allows you to make customizations to Oracle Knowledge.

By selecting **Custom Configuration, Show Advanced Options, Edit** the Editing: Custom Configuration screen displays.

Configuration

Lists customizations made to Oracle Knowledge. To modify select **Add New Item**.

E-mail Notification

Selecting E-mail Notification from Advanced Configuration Facility allows you to see and modify to e-mail notifications.

By selecting **E-mail Notification, Show Advanced Options, Edit** the Editing: E-mail Notification screen displays.

E-mail From

Refers to which Oracle Knowledge Instance e-mails are sent from. The default value is scheduler.

E-mail Host	Specifies the name of the mail server accessible by the server used for Oracle Knowledge Instance defined above
E-mail Group	Lists the defined e-mail groups to be notified. To modify select Add New Item .

Workbench

Selecting Workbench from Advanced Configuration Facility allows you to see and modify components related to the Language Workbench in Oracle Knowledge Intelligent Search.

By selecting **Workbench, Show Advanced Options, Edit** the Editing: Workbench screen displays.

Datasource (required)	Lists the datasource configured for Intelligent Search. To modify select Edit List .
Root Concepts	Specifies one or more concepts as initial display nodes for the Dictionary Manager Concept Tree. To modify select Add New Item .
Browser Executable (Unix Only)	Specifies the browser to use for Workbench features, such as Test Drive, Quality Monitor, Process Wizards. The default value is firefox.
Browser Arguments (Unix Only)	This argument corresponds to the defined browser above. The default value is -remote openURL.

Information Manager

Selecting Information Manager from Advanced Configuration Facility allows you to see and modify the log directory used for Oracle Knowledge Information Manager.

By selecting **Information Manager, Show Advanced Options, Edit** the Editing: info-manager screen displays.

Datasource	Specifies the Datasource to use when accessing Oracle Knowledge Information Manager.
Information Manager Log Files Directories	Specifies the directory used to store Information Manager log files. To modify, select Add New Item .
User	n/a
Password	n/a
Repository	n/a
URL	n/a

Rules Engine

Selecting Rules Engine from Advanced Configuration Facility allows you to see and modify components related to the question processing used by Oracle Knowledge Intelligent Search.

By selecting **Rules Engine, Show Advanced Options, Edit** the Editing: Rules Engine screen displays.

The screenshot shows the Oracle Advanced Configuration interface. On the left is a navigation menu with categories like System, Instances, and Tools. The main area is titled 'Rules Engine' and contains several configuration sections:

- Rules Engine:**
 - Max Question Length (in tokens) : 15
 - Truncated Question Length (in tokens) : 14
 - Use First Tokens : Off
- Spell Checking:**
 - Automated : On
 - Interactive : Off
 - Automatic Correction Threshold : 90
 - Don't spell-check words in the question that are found in my content : On
 - How many times must the word appear? : 4
 - Do not look for words in these collections : (none)

At the bottom of the configuration area are two red buttons: 'Save' and 'Edit'.

Max Question Length (in tokens) Specifies the maximum amount of tokens to be analyzed by Intelligent Search in the answer retrieval process.

Truncated Question Length (in tokens) Specifies how many tokens to truncate a question to if the question exceeds the Max Question Length.

Use first token Specifies whether to truncate the question to the set max length starting from the beginning of the question. The default is off

Spell Checking

Automated Specifies whether automated spelling checking is used to process a question. The default value is **On**. Setting this to **Off** removes the automatic mode of operation of the Spell-Checker.

Interactive Specifies whether to automatically suggest spelling corrections. The default value is off. Setting this to **On** while the Automated setting is **Off** enables the interactive mode. Setting both interactive and automated to **Off** turns off the spell-checking functionality altogether.

Automatic Correction Threshold Specifies a threshold as a percentage below which suggested alternate words are not submitted to the question answering process. The recommended default is 90. This means that corrections are applied when the system's internal confidence that the correction is appropriate is at least 90%. Lowering this threshold produces more corrections, but it increases the possibility of over-corrections. In automated mode, over-corrections can have a negative impact on search accuracy.

Don't spell check words in the question that are found in my content	This setting turns On and Off the learning-from-content functionality.
How many times must the word appear?	This sets a minimum frequency threshold for a word found in the content to be considered a legitimate, correctly spelled word. It is provided to protect against accidental misspellings found in the content, for example when portions of the content are unedited, as can be the case with Forum and Discussion Boards. The default threshold (=4) can be lowered when all the content has been edited.
Do not look for words in these collections	When the content contains collections that are unedited and known to include many misspellings, the Spell-Checker can be instructed not to learn words from these collections.

Language Processor Settings

Selecting Language Processor Settings from Advanced Configuration Facility allows you to see and modify components related to natural language settings used by Oracle Knowledge Intelligent Search.

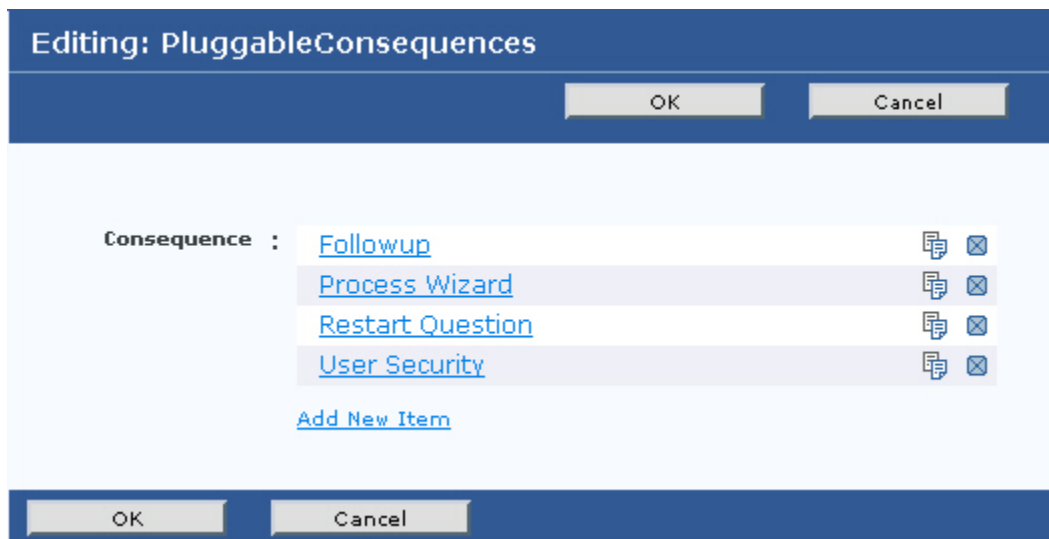
By selecting **Language Processor Settings, Show Advanced Options, Edit** the Editing: Natural Language Settings screen displays.

Use Link as References	Allows you to search and return answers based on tokens within href links. The default value is off.
OpenClass (Index)	Refers to whether documents get annotated with openclass concepts. The default value is off.
OpenClass (Runtime)	Refers to whether questions are annotated with openclass concepts. The default value is on.
openclass-item	Allows you to create and modify openclass concepts. An openclass concept is a concept defined using IML. To modify select Add New Item .

Plug-in Registration

Selecting Plug-in Registration from Advanced Configuration Facility allows you to see and modify the plug-ins associated with Oracle Knowledge.

By selecting **Plug-in Registration, Show Advanced Options, Edit** the Editing: PluggableConsequences screen displays.



Consequence Allows you to configure code to invoke various plug-ins as a result of language rule matching. To modify select **Add New Item**.

Preprocess Configuration

Selecting Process Config from Advanced Configuration Facility allows you to see and modify PDF and HTML settings.

By selecting **Preprocess Config, Show Advanced Options, Edit** the Editing: Preprocess Config screen displays.

Title in First Page	n/a The default value is true.
Max Sentences	Refers to the maximum number of sentences to contain the document title. The default value is 5
Max Title Length	Refers to the maximum number of characters to be included in a PDF title. The default value is 100.
Timeout Factor	Specifies how many ms/KB to be used in the algorithm to avoid PDFs from hanging. The default value is 3.35
Timeout Contingency	Specifies how many ms to be used in the algorithm to avoid PDFS from hanging. The default value is 15000
PDF Rejection Pattern	Refers to regular expressions used to reject potential PDF titles

PDF Title Candidate	Lists ways, in numerical order, Oracle Knowledge finds titles in PDF documents. To modify select Add New Item .
Notify HTML Built in Refs	Specifies whether to use the HTML parser features to automatically report character entity references. The default value is off.
Fix MS Windows Refs	Specifies whether to use the HTML parser features to automatically fix Microsoft Windows characters. The default value is off.
Notify Char Refs	Specifies whether to use the HTML parser features to automatically report character entity references. The default value is off.
Document Title	Specifies an HTML style to display an HTML document title.
Title for Start section	Specifies an HTML style to create a new section in an HTML document.
Keep Entity	Specifies a retained node in order to look at data and extract text. To modify select Add New Item .
Directive Rule	Defines HTML nodes. To modify select Add New Item .

Instances

The Instances portion of the Advanced Configuration Facility provides a means to modify setting that affect a specific instance of Oracle Knowledge. For more information see the following sections:

[Communication Settings on page 147](#)

[Gateways on page 148](#)

[Instance Groups on page 149](#)

[Synchronizer Defaults on page 150](#)

[Service Configuration on page 151](#)

Communication Settings

Selecting Communication Settings from Advanced Configuration Facility allows you to see and modify the means of communication used by Oracle Knowledge.

By selecting **Communication Settings**, **Show Advanced Options**, **Edit** the Editing: Communication Methods screen displays.

Editing: Communication Methods

RMI Communication : (none)
[Add New Item](#)

EJB Communication : (none)
[Add New Item](#)

SOAP Communication : (none)
[Add New Item](#)

Socket Communication :

default Socket Transport	<input type="checkbox"/> <input type="checkbox"/>
MLD24rt1 Socket Transport	<input type="checkbox"/> <input type="checkbox"/>
scheduler Socket Transport	<input type="checkbox"/> <input type="checkbox"/>

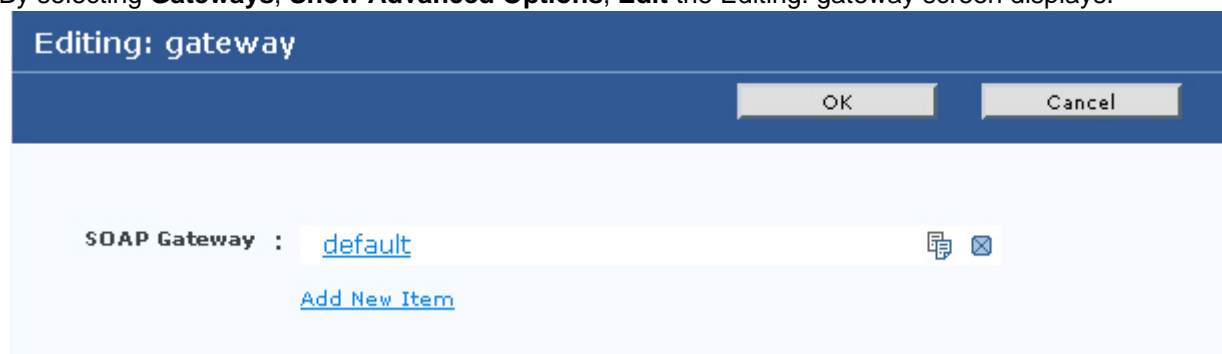
[Add New Item](#)

EJB Communication	Lists the available Enterprise Java Bean configurations for the instance. To modify select Add New Item .
SOAP Communication	Lists the available Simple Object Access Protocol configurations for the instance. To modify select Add New Item .
Socket Communication	Lists the available socket configurations for the instance. To modify select Add New Item .

Gateways

Selecting Gateways from Advanced Configuration Facility allows you to see and modify the Gateway connections used by Oracle Knowledge.

By selecting **Gateways**, **Show Advanced Options**, **Edit** the Editing: gateway screen displays.

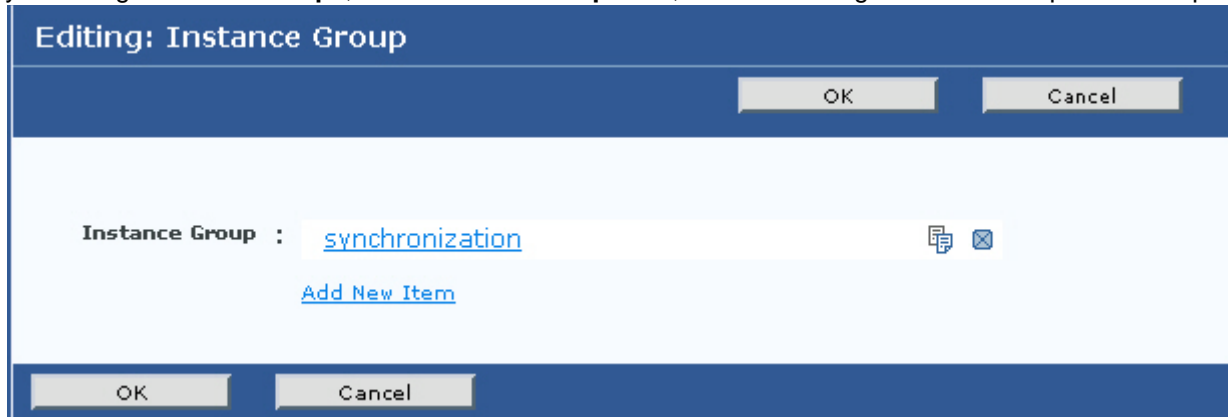


SOAP Gateway	Lists configured SOAP Gateway connections. To modify select Add New Item
---------------------	---

Instance Groups

Selecting Instance Groups from Advanced Configuration Facility allows you to see and modify groups of Oracle Knowledge instances.

By selecting **Instance Groups, Show Advanced Options, Edit** the Editing: Instance Group screen displays:



Instance Group

Lists Oracle Knowledge runtime instances that have been grouped together to perform a set task. To modify select **Add New Item**.

Synchronizer Defaults

Selecting Synchronizer Defaults from Advanced Configuration Facility allows you to see and modify n/a

By selecting Synchronizer Defaults, **Show Advanced Options**, **Edit** the Editing: Synchronizer Defaults screen displays:

Instance Group

Lists the defined instance groups for the Oracle Knowledge application. To modify select **Edit List**.

Target Application

Lists the available Oracle Knowledge applications. To modify select **Edit List**.

Instance Failure Threshold

Defines the number of synch tasks a may fail before it is considered to be an error. The default value is 0.

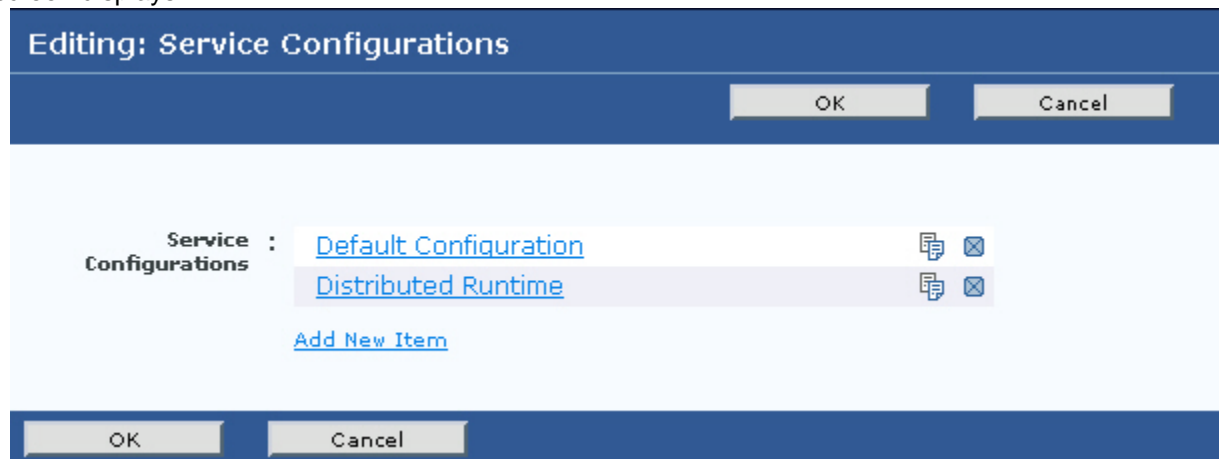
Development Data Cleanup History

Specifies to keep data for the last X number of runtime revisions, while removing all older data after a successful synch.

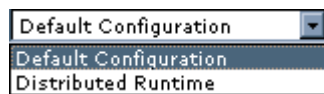
Service Configuration

Selecting Service Configurations from Advanced Configuration Facility allows you to see and modify configured Oracle Knowledge Instances.

By selecting **Service Configurations**, **Show Advanced Options**, **Edit** the Editing: Service Configurations screen displays:



Service Configurations Lists the available Oracle Knowledge instances. To modify select **Add New Item**.



From the drop down menu you can see and modify settings specific to that instance.

For more information see:

[Content Storage on page 151](#)

[Context Variable on page 153](#)

[Default-Subjectmap on page 154](#)

[Excerpt on page 155](#)

[Excerpt Security on page 156](#)

[Index on page 157](#)

[Log on page 160](#)

[Ontology on page 160](#)

[Structured Data on page 161](#)

[Authorization on page 162](#)

Content Storage

Selecting Content Storage from Advanced Config allows you to see and modify settings related to Oracle Knowledge Intelligent Search.

By selecting **Content Storage, Show Advanced Options, Edit** the Editing: Content Storage screen displays.

Editing: Content Storage

Show Advanced Options

Data Source : [Edit List](#)

Default Encoding ▶

Default Language ▶

Advanced Options

Reset Commit Interval :

- | | |
|-------------------------------|---|
| Data Source (required) | Lists the database associated with Oracle Knowledge Intelligent Search content store. To modify select Edit List . |
| Default Encoding | Specifies the encoding used during crawling. |
| Default Language | Specifies the default language of the documents to be crawled. |
| Reset Commit Interval | Refers to the number of documents that are deleted before a database commit is done. The default value is 2. |

Context Variable

Selecting Context Variables from Advanced Config allows you to see and modify the context variables used by Oracle Knowledge Intelligent Search.

By selecting **Context Variables**, **Show Advanced Options**, **Edit** the Editing: Context Variables screen displays.



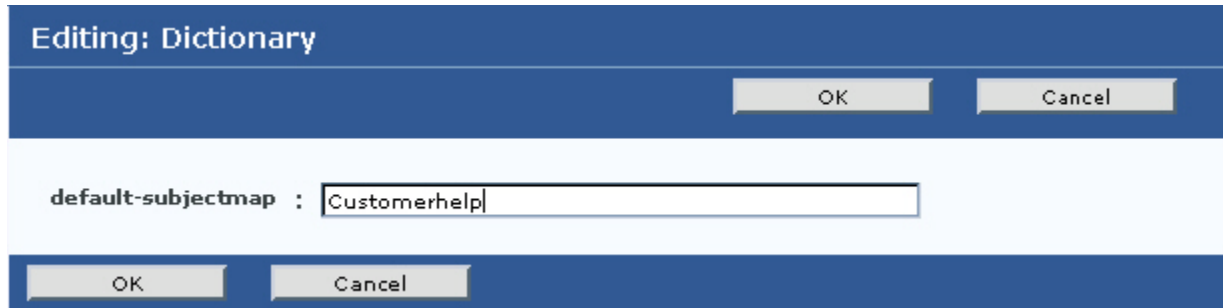
Variables

Lists the variables used within Oracle Knowledge. To modify select **Edit List**.

Default-Subjectmap

Selecting Default-Subjectmap from Advanced Config allows you to see and modify the default dictionary used by Oracle Knowledge Intelligent Search

By selecting **Default-Subjectmap, Show Advanced Options, Edit** the Editing: Dictionary Services screen displays.



The screenshot shows a dialog box titled "Editing: Dictionary". It has a blue header bar with the title. Below the header, there are two buttons: "OK" and "Cancel". The main content area is white and contains the text "default-subjectmap : " followed by a text input field containing the value "Customerhelp". At the bottom of the dialog, there are two more buttons: "OK" and "Cancel".

default-subjectmap

Specifies the default subject map to use with Oracle Knowledge Intelligent Search.

Excerpt

Selecting Excerpt from Advanced Config allows you to set and modify excerpt parameters.

By selecting **Excerpt, Show Advanced Options, Edit** the Editing: Excerpt Service screen displays.

Answer Format Algorithms

Lists the available portlets. To modify select **Add New Item**.

Maximum Open Files

Refers to the maximum number of open files to be used by Oracle Knowledge. The default value is 512

You can have an excerpt even for documents that only matched the question via the document title. This allows you to include context sentences for titles.

To add excerpts to title matches, modify the `titleWindow` item for the following configuration options to be greater than 0:

```
<excerptAlgorithm index="4">
  <name>Sentence Scope Algorithm</name>
  <class>com.InQira.excerpt.SentenceScopeExcerptAlgorithm</class>
  <parameter name="window">3</parameter>
  <parameter name="titleWindow">0</parameter>
  <parameter name="max-tokens">300</parameter>
</excerptAlgorithm>
<excerptAlgorithm index="5">
```

```

    <name>Small Excerpt Algorithm</name>
    <class>com.InQuira.excerpt.SentenceScopeExcerptAlgorithm</class>
    <parameter name="window">1</parameter>
    <parameter name="max-tokens">20</parameter>
    <parameter name="titleWindow">0</parameter>
  </excerptAlgorithm>
  <excerptAlgorithm index="6">
    <name>Document/Section Scope Algorithm</name>
    <class>com.InQuira.excerpt.TransSentenceScopeExcerptAlgorithm</class>
    <parameter name="max-tokens">300</parameter>
    <parameter name="titleWindow">0</parameter>
  </excerptAlgorithm>

```

Excerpt Security

Selecting Excerpt Security from Advanced Config allows you to see and modify n/a

By selecting Excerpt Security, Show Advanced Options, Edit the Editing: Excerpt Security Service screen displays.

Enable Excerpt Security	N/A. The default value is off
LDAP Security	Lists the available methods used to communicate with the LDAP security. To modify select Edit List .
Security Class	The code used to communicate with LDAP security

Index

Selecting Index from Advanced Config allows you to see and modify settings related to the Oracle Knowledge Intelligent Search results.

By selecting **Index, Show Advanced Options, Edit** the Editing: Index Service screen displays.

Editing: Index Service

Show Advanced Options OK Cancel

Directory ▾ data/index

Parameters

Default Search Weight : 0.99999999

Default Relevance Weight : 0.00000001

Default Recency Weight : 0

Maximum # of Search Results : 15

Maximum # of Search Results per Document : 1

dynamic-SCCache-enabled : On Off

uncomment-enabled : On Off

guarantee-max-results-enabled : On Off

use-delayed-doc : On Off

max-delayed-docs : 1000

max-delayed-tied : 10

use-answer-buckets : On Off

max-answers-per-bucket : 4

max-memory : 786000

io-buffer-size : 32

Cache Sizes : query

Directory

Refers to the directory used to store data related to the Index. The default is data/index

Default Search Weight	Refers to the weight given to a document based on created dictionary rules. The default value is .99999999
Default Relevance Weight	Refers to the weight given to a document based on how often that document is linked to other documents. The default value is .00000001
Default Recency Weight	Refers to the weight given to documents based on the frequency of use. The default value is 0.
Maximum # of Search Results	Refers to the maximum number of search results returned for any question. The default value is 15. (Note: This configuration takes precedence over the max results setting defined under Dictionary.)
Maximum # of Search Results per Document	Refers to the maximum number of times one article may be returned in the search results. The default value is 1.
dynamic-SCCache-enabled	Specifies whether to update the Search Component cache is realtime, allowing answers to be returned faster. The default value is on.
uncomment-enabled	Enables very low level scoring debug. The default value is off.
guarantee-max-results-enabled	Specifies to return the configured maximum number of results regardless of scores. The default value is off.
use-delayed-doc	n/a The default value is off.
max-delayed-docs	n/a The default value is 1000.
max-delayed-tied	Alternate scoring algorithm that is useful only if document weighting is high. The default value 10.
use-answer-buckets	n/a The default value is off.
max-answers-per-bucket	Allows you to set a maximum number of documents that can be returned as answers per "answer bucket." The default value is 4.
max-memory	Refers to the maximum amount of memory to allocate to the search User Interface. The default value is 786000.
io-buffer size	Changes the size of the IO buffer used by the UDS. The default value is 32.
Cache Sizes	Specifies the amount of memory allowed for caching IML results. To modify select Add New Item .

Relevance Page Rank Weight	Specifies how relevant is page ranking. The default value is .9. The value must be set between 0 and 1, and in combination with the TFIDF (term frequency–inverse document frequency) Weight may not equal greater than 1.
Relevance TFIDF Weight	Specifies the relevance of the TFIDF (term frequency–inverse document frequency) score. The default value is .1. The value must be set between 0 and 1, and in combination with the Page Rank Weight may not equal greater than 1.
Title Boost	Specifies the amount of points to assign to a result appearing in a document title. The default value is 10.
Reoccurrence in Sentence Boost	Specifies the amount of points to assign to a result appearing numerous times in a document sentence. The default value is 0.
Reoccurrence in Section Boost	Specifies the amount of points to assign to a result appearing numerous times in a document section. The default value is 0.
Title Boos Max	Specifies the maximum amount of points to assign to a result appearing in a document title.
tf-ruleid-keyword	Specifies the value used in the TFIDF algorithm. The default value is 1755852687.
tf-ruleid-nl	Specifies the value used in the TFIDF algorithm. The default value is 799415718.
tf-slot	Specifies the value used in the TFIDF algorithm. The default value is 7.
base-score-floor	Specifies the value used in the TFIDF algorithm. The default value is .7.
industry-score-floor	Specifies the value used in the TFIDF algorithm. The default value is .4.
Heuristic Always Optimal-pos	N/A. The default value is 3.
Heuristic Negligible-score-changes	N/A. The default value is 0.
Scale Final Score	N/A. The default value is off.
Score Scaling Exponent	N/A. The default value is 5.
Score Scaling Base	N/A. The default value is 10.
Scope Limit Per Document	N/A. The default value is 0.
Preload SC Cache	N/A. The default value is off.

Preload Document Structure Cache N/A. The default value is off.

Log

Selecting Log from Advanced Config allows you to see and modify the log files related to Oracle Knowledge Analytics.

By selecting **Log, Show Advanced Options, Edit** the Editing: Log Service screen displays.

Verbosity	Refers to the amount of detail reported within log files
mode	N/A.
Log Directory	Specifies the directory used to store log files. The default value is logs.

Ontology

Selecting Ontology from Advanced Config allows you to see and modify settings regarding the n/a

By selecting **Ontology, Show Advanced Options, Edit** the Editing: Ontology Service screen displays.

Structured Data

Selecting Structured Data from Advanced Config allows you to see and modify the databases configured. By selecting **Structured Data, Show Advanced Options, Edit** the Editing: Structured Data Service screen displays.

Editing: Structured Data Service

Show Advanced Options OK Cancel

Schemas

Table : (none)
[Add New Item](#)

Options

Max Answers : 500

Max Ranking Answers : 30

Max IML Results In Query : 500

Process Subselects : On Off

XML Query Script :

XML Results Script :

Advanced Options

Directory ▶ data/structured

Connection Type ▶ idbc
xml

Table Lists the tables within the Structured Database. To modify select **Add New Item**.

Max Answers Refers to the maximum number of allowed results returned from a structured query. The default value is 500.

Max Ranking Answers	Refers to the maximum number of allowed results from an unstructured search of a structured database. The value should be less than the Max Answers configured above. The default value is 30.
Max IML Results In Query	Refers to the maximum number of allowed results from an IML query to be analyzed in a structured query. The default value is 500.
Process Subselects	Specifies whether to allow Oracle Knowledge to process subselect statements in the SQL query, regardless of the capabilities of the associated structured database. The default value is on
XML Query Scrip	Allows structured search using XML not SQL in question processing.
XML Results Script Directory	Allows structured search using SML not SQL in search retrieval.
Directory	Specifies the directory to used to store files from the structured database.
Connection Type	Specifies the method used to connect to the structured database. To modify select Add New Item .

Authorization

Selecting Authorization from Advanced Config allows you to see and modify n/a

By selecting **Authorization**, **Show Advanced Options**, **Edit** the Editing: Authorization screen displays.

Editing: Authorization

Show Advanced Options OK Cancel

Authentication Domain : 1. [internal](#) [Copy] [X]

[Add New Item](#)

Roles : [Administrator](#) [Copy] [X]

[LanguageAdministrator](#) [Copy] [X]

[LanguageDevelopment](#) [Copy] [X]

[Add New Item](#)

Advanced Options

Delegation Detector : (none) [Edit List](#)

OK Cancel

Authentication Domain Lists available LDAP domains. To modify select **Add New Item**.

Roles Lists available user roles. To modify select **Add New Item**.

Delegation Detector

Used to integrate with Single-Sign-On solutions (SSO). When configured, if there is no user associated with the current session Oracle Knowledge attempts to detect if a security delegation has occurred. To modify select **Edit List**.

