Oracle® Secure Enterprise Search

Administrator's Guide

10*g* Release 1 (10.1.7) Beta

B32011-01

August 2006

Beta Draft

ORACLE®

Oracle Secure Enterprise Search Administrator's Guide, 10*g* Release 1 (10.1.7) Beta

B32011-01

Primary Author: Michele Cyran

Contributors: Omar Alonso, Edwin Balthes, Sachin Bhatkar, Meeten Bhavsar, Stefan Buchta, Thomas Chang, Mark Davis, Sudhir Dureja, Roger Ford, Cindy Hsin, Diego Iglesias, Hiroshi Koide, Vishu Krishnamurthy, Muralidhar Krishnaprasad, Ciya Liao, Jun Miao, Tommy Mo, Arup Mohanty, Valarie Moore, Huyen Nguyen, Visar Nimani, Hui Ouyang, Rakesh Patel, Janaka Ranatunga, Yi Tan, Jenny Tsai, Mark Ture, Madhu Velukur, Luke Wang, Xiaofang Wang, Steve Yang, Ying Yu

Contents

4 Security in Oracle Secure Enterprise Search

5 Oracle Secure Enterprise Search Advanced Information

6 Oracle Secure Enterprise Search APIs

Preface

This Preface contains these topics:

> **See Also:** *Oracle Secure Enterprise Search Release Notes* for version information and known issues, and *Oracle Secure Enterprise Search Installation and Upgrade Guide for Linux x86* for preinstallation requirements, installation tips, and information on how to get started using Oracle Secure Enterprise Search

Audience

Oracle Secure Enterprise Search Administrator's Guide is intended for administrators and application developers who perform the following tasks:

- Install and configure Oracle Secure Enterprise Search
- Administer Oracle Secure Enterprise Search
- Develop Oracle Secure Enterprise Search applications

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Conventions

The following text conventions are used in this document:

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

What's New

This chapter describes new features of Oracle Secure Enterprise Search 10g Release 1 (10.1.7) and provides pointers to additional information.

New Features for Release 10.1.7

Oracle Secure Enterprise Search 10g Release 1 (10.1.7) is a limited Beta release, available to selected customers only. It will be upgradable to version 10.1.8.

Release 10.1.7 contains the following new features:

- Built-in source types for Oracle Calendar and Oracle Content Database/Oracle Content Services (Oracle Content Database and Oracle Content Services are the same product)

  See Also: "Setting Up Secure Oracle Content Database Sources" on page 5-8 and "Setting Up Secure Oracle Calendar Sources" on page 5-6

- Simplified configuration for federated sources

  See Also: "Tips for Using Federated Sources" on page 5-4 and "Setting Up Secure Federated Search" on page 5-4

- Ability to register any identity management system for validating and authenticating users (that is, there is no longer a dependency on Oracle Internet Directory for search)

  See Also: "Authorization and Authentication" on page 4-2

- Ability to override the default document title with a meaningful title if the default title is deemed irrelevant

  See Also: "Title Fallback" on page 3-5

1

Introduction to Oracle Secure Enterprise Search

This chapter contains the following topics:

Overview of Oracle Secure Enterprise Search

Oracle Secure Enterprise Search (SES) provides uniform search capabilities over multiple repositories.

Oracle SES uses a crawler to collect data from these sources. The crawler supports a number of built-in source types, as well as a published plug-in (or *connector*) architecture for adding new types. Multiple Oracle SES instances can also share content through the federated source type.

Oracle SES supports the following built-in source types:

- **Web**: A Web source represents the content on a specific Web site. Web sources facilitate maintenance crawling of specific Web sites.
- **Table**: A table source represents content in an Oracle database table or view.
- **File**: A file source is the set of documents that can be accessed through the file protocol.
- **E-mail**: An e-mail source derives its content from e-mails sent to a specific e-mail address. When Oracle SES crawls an e-mail source, it collects e-mail from all folders set up in the e-mail account, including Drafts, Sent Items, and Trash e-mails.
- **Mailing list**: A mailing list source derives its content from e-mails sent to a specific mailing list.
- **OracleAS Portal**: An OracleAS Portal source allows users to search across multiple OracleAS Portal repositories, such as Web pages, files on disk, and pages on other OracleAS Portal instances.
- **Federated**: A federated source represents a connection to a remote Oracle SES instance or application that maintains its own index. Oracle SES can issue a search to this remote instance, and the remote instance can return results.
- **Oracle Calendar**: An Oracle Calendar source represents the content in an Oracle Calendar repository. Oracle SES can crawl content (meetings and events) and

metadata in Oracle Calendar and provide secure full-text search over an Oracle Calendar repository. You can specify more than one thread to crawl. Deleted items are removed from the index during incremental crawling. You can search based on title, author, start or end date (year, month, day), event type, status, or location.

- **Oracle Content Database**: An Oracle Content Database source represents the content in an Oracle Content Database repository.

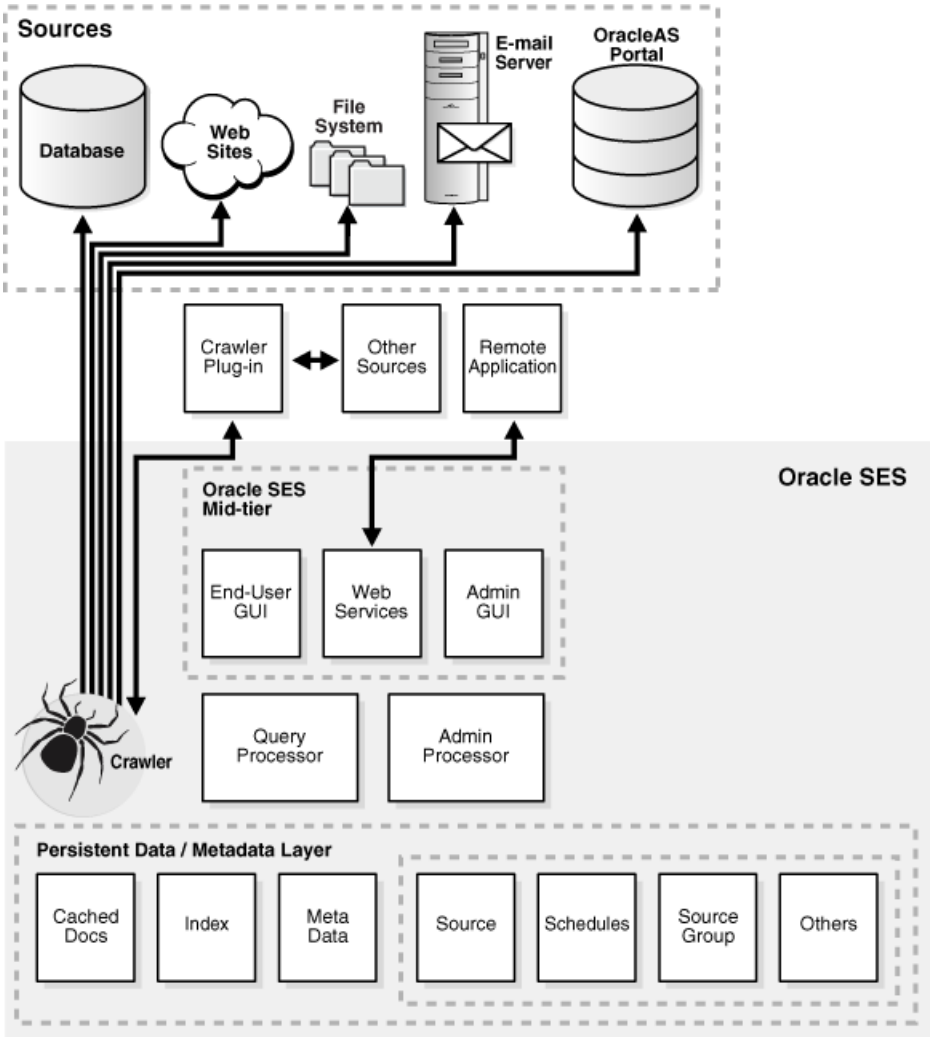
  Note: Oracle Content Database and Oracle Content Services are the same product. This book uses the product name Oracle Content Database to mean Oracle Content Database *and* Oracle Content Services. Oracle Content Database sources are certified with Oracle Content Database release 10.2 and Oracle Content Services release 10.1.2.3.

  Oracle SES can crawl documents and metadata in Oracle Content Database and provide secure full-text search over an Oracle Content Database repository. It also provides metadata search and browse, which allows a search to be done against a specific subfolder in the hierarchy. Documents in Oracle Content Database are organized into *Folders*. Oracle SES navigates the folder hierarchy to crawl all documents in Oracle Content Database. It creates an index, stores the metadata, and accesses information in Oracle SES to provide search according to the end users' permissions.

  Oracle SES supports incremental crawling; that is, it only crawls and indexes documents that have changed since the last crawling. A document is re-crawled if either the content or the direct security access information of the document changes. A document is also re-crawled if it is moved within Oracle Content Database and the end user has to access the same document with a different URL. Deleted documents are removed from the index during incremental crawling.

  See Also: "Setting Up Secure Oracle Calendar Sources" on page 5-6 and "Setting Up Secure Oracle Content Database Sources" on page 5-8

The following diagram illustrates Oracle SES architecture.



See Also:

- *Oracle Secure Enterprise Search Release Notes* for version information and known issues
- *Oracle Secure Enterprise Search Installation and Upgrade Guide for Linux x86* for installation requirements and tips, upgrade steps, and information on how to get started using Oracle SES
- The Oracle SES home page for updated information on known issues, as well as code samples and best practices: `http://www.oracle.com/technology/products/oses/index.html`

Oracle Secure Enterprise Search Components

Oracle SES includes the following components:

- Oracle Secure Enterprise Search Crawler
- Oracle Secure Enterprise Search Administration Tool
- Oracle Secure Enterprise Search APIs and Applications

Oracle Secure Enterprise Search Crawler

The Oracle SES crawler is a Java process activated by a set schedule. When activated, the crawler spawns a configurable number of processor threads that fetch information from various sources and index the documents. This **index** is used for searching **sources**.

The crawler maps links and analyzes relationships. Whenever the crawler encounters embedded non-HTML, or non-textual documents during the crawling, it automatically detects the document type and filters and indexes the document.

See Also: Chapter 3, "Understanding Crawling and Searching"

Oracle Secure Enterprise Search Administration Tool

Use the Oracle Secure Enterprise Search administration tool to manage and monitor Oracle SES components. For example:

- Define sources and crawling scope
- Configure the search application
- Monitor crawl progress and search performance

See Also:

- "Understanding the Administration Tool" on page 2-2
- Oracle SES admin tutorial for help understanding common administrator tasks:

  `http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm`
- Oracle SES administration tool context-sensitive online help

Oracle Secure Enterprise Search APIs and Applications

Oracle Secure Enterprise Search provides several APIs. For example, the Crawler Plug-in API enables you to create a custom secure crawler plug-in (or *connector*) to meet your requirements. With the Web Services API, you can integrate Oracle SES search capabilities into your search application.

Oracle SES also provides an out-of-the-box search application.

See Also:

- Chapter 6, "Oracle Secure Enterprise Search APIs"
- *Oracle Secure Enterprise Search Java API Reference*

Oracle Secure Enterprise Search Features

Information in an enterprise can be spread across Web pages, databases, mail servers or other collaboration software, document repositories, file servers, and desktops. Oracle SES searches all your data through the same interface. Oracle SES is fully globalized and works with 27 languages including Chinese, Japanese, Korean, Arabic, and Hebrew.

This section introduces a few of the features in Oracle SES. It includes the following topics:

- Secure Search
- Federated Search
- Web Services API
- Extensible Crawler Plug-in Framework

> **See Also:** Chapter 3, "Understanding Crawling and Searching" for more features relating to the crawler

Secure Search

Much of the information within an organization is publicly accessible. Anyone is allowed to view it. Therefore, it is relatively easy for a **crawler** to find and index that information.

However, there are other sources that are protected. These protected sources might only be viewable by certain users or groups of users. For example, while users can search within their own e-mail folders, they should not be able to search anyone else's e-mail.

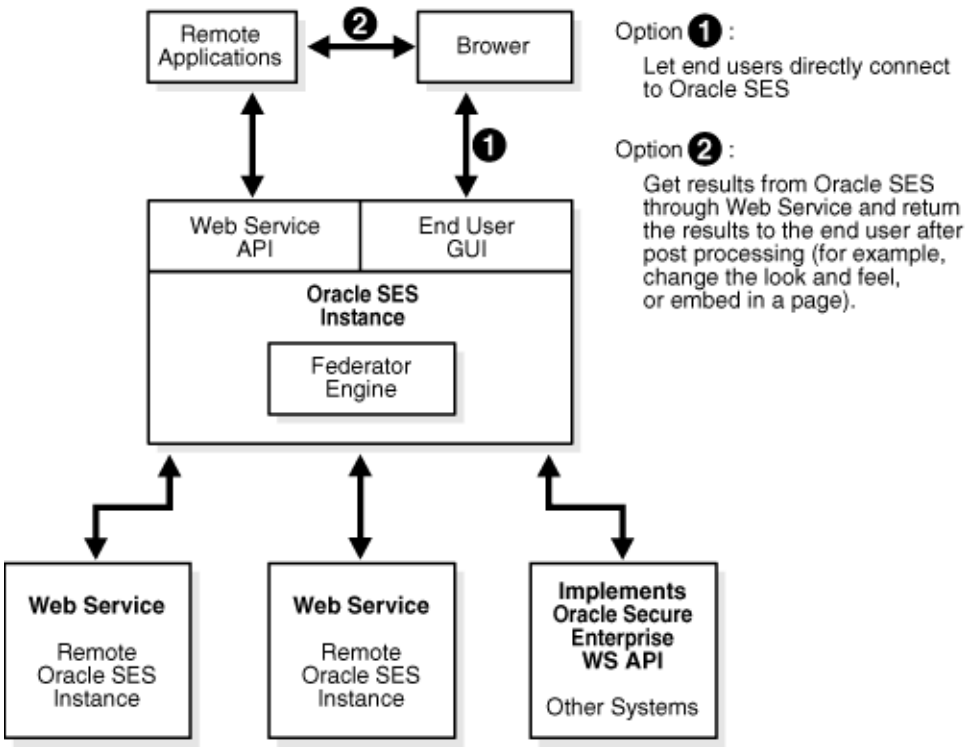
For protected sources, the Oracle SES crawler will index documents with the proper access control list. When end users perform a search, only documents that they have privileges to view will be returned.

> **See Also:** "Enabling Secure Search" on page 4-7

Federated Search

Oracle Secure Enterprise Search provides the capability of searching multiple Oracle SES applications with their own document repositories and indexes. It provides a unified framework to search the different document repositories that are crawled, indexed, and maintained separately. Federated search allows a single query to be run across all indexes. It aggregates the search results to show one result list to the user. User credentials are passed along with the search so that each remote application can authenticate the user against its own document repository.

The following diagram illustrates Oracle SES federation architecture.



Web Services API

Oracle SES offers a Web services API that lets you integrate Oracle SES search capabilities into your search application.

> **See Also:** "Oracle Secure Enterprise Search Web Services API" on page 6-1

Extensible Crawler Plug-in Framework

Oracle SES provides an extensible crawler plug-in (or *connector*) framework that lets you crawl and index proprietary document repositories.

> **See Also:**
>
> - "Crawler Plug-in API" on page 6-25
> - The Oracle Secure Enterprise Search home page at `http://www.oracle.com/technology/products/oses/index.html` for updated information on known issues, as well as code samples and best practices

2

Getting Started with Oracle Secure Enterprise Search

This chapter provides a brief introduction to using Oracle Secure Enterprise Search. More information is provided later in this book, as well as in the online help for the administration tool.

This chapter contains the following topics:

Getting Started Basics with Oracle Secure Enterprise Search

After you have successfully installed Oracle SES, you can start crawling your data. Open a browser, enter the URL provided at the end of the installation for the administration tool (`http://host:port/search/admin/index.jsp`), and log on.

Here are the basic steps to start using Oracle SES quickly:

1. Define one or more sources for the data you want to search on the **Home - Sources** page. For example, if your data is in Web pages, then select Web source. A crawl schedule is automatically created along with the source. If **Start Crawling Immediately** is selected, then the crawler will start crawling after you click **Create**.
2. Check the crawler progress and status on the **Home - Schedules** page. (Click **Refresh Status**.) From the status page, you can view statistics of the crawl.
3. Test whether users can search this source by clicking the **Search** link in the upper right corner of any page. This brings up the search page in a new window. (The URL for **Search** should be `http://host:port/search/query/search`.
4. Monitor your search statistics on the **Home - General** page and the **Home - Statistics** page.

Note: By default, Oracle SES is configured to crawl Web sites in the intranet. To crawl Web sites on the Internet (also referred to as external Web sites), Oracle SES needs the HTTP proxy server information. See the **Global Settings - Proxy Settings** page.

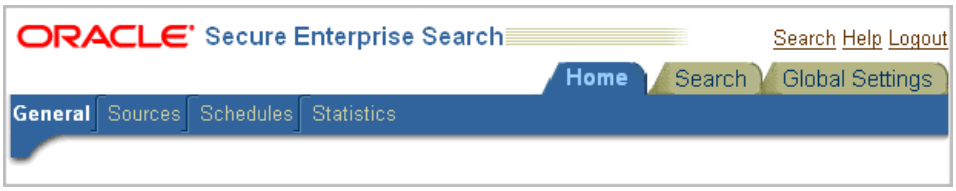
You might also want to define crawling parameters before you start crawling.

Understanding the Administration Tool

There are many options in the administration tool for managing and customizing Oracle SES to suit your enterprise. This section describes some of the tasks available in the administration tool.

Home Tab

The **Home** tab consists of the **General**, **Sources**, **Schedules**, and **Statistics** subtabs.



- **Home - General Page**

  This is the home page for Oracle SES. The **Summary** section shows an overview of the system's search performance, both quality and speed, over the past seven days. The **Failed Schedules** section lists all schedules that have failed. Generally, a failed schedule is one in which the crawler did not collect any documents. A failed schedule also could be the result of a partial collection and indexing of documents.

- **Home - Sources Page**

  A collection of information is called a source. Each source has a type, such as a Web site or a database table. Sources can be Web sites, database tables, files, e-mail, mailing lists, OracleAS Portal page groups, federated sources, Oracle Calendar repositories, Oracle Content Database/Oracle Content Services repositories, or user-defined sources.

  User-defined source types are created on the **Global Settings - Source Types Page**. The drop-down list includes any available user-defined source types. You can create as many sources as you want.

- **Home - Schedules Page**

  This page lets you view, edit, create, delete, stop, or start a schedule. Schedules define the frequency at which the index is updated with information about each source.

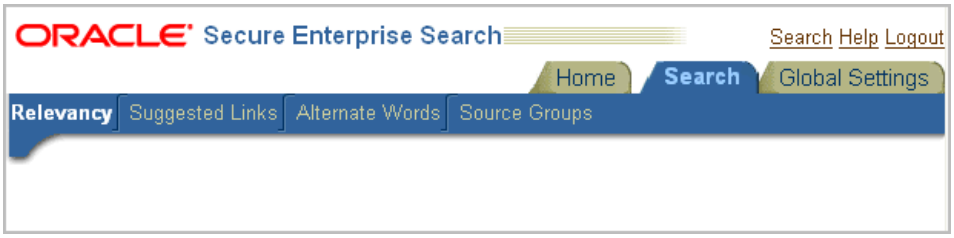
- **Home - Statistics Page**

  This page provides numerous search and crawler statistics, such as most popular queries and crawler progress.

  Note: Some statistics constantly show up-to-date information, while others are cached hourly to improve performance. The **Last Refreshed** time shows the actual time of the statistics displayed. Check the online help for each statistics page to confirm if the statistics are up-to-date or cached hourly.

Search Tab

The **Search** tab consists of the **Relevancy**, **Suggested Links**, **Alternate Words**, and **Source Groups** subtabs. These pages help you improve search performance.



- **Search - Relevancy Page**

  Make important documents easier to find with relevancy boosting. Oracle SES lets you influence the order of documents in the result list for a particular search. For example, your company Web site could have a home page for documentation that you want to appear high in the results of any search for "documentation".

- **Search - Suggested Links Page**

  Direct users to a particular Web site for a search string. For example, when users search for "Oracle SES documentation" or "Enterprise Search documentation" or "Search documentation", you could suggest http://www.oracle.com/technology. In the default search page, suggested links are displayed at the top of the search result list. This is especially useful to provide links to important Web pages that are not crawled by Oracle SES.

- **Search - Alternate Words Page**

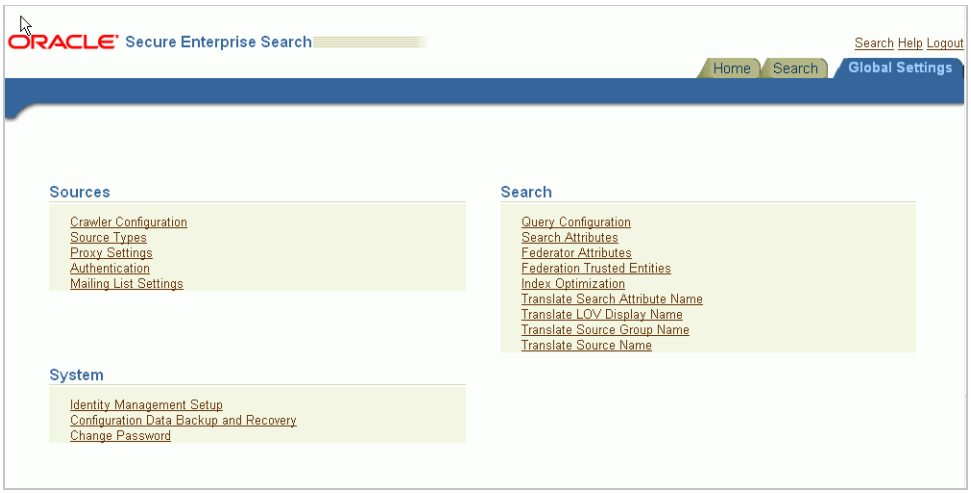
  Use alternate words to suggest alternative search queries to users. This is useful for fixing common errors that users make in searching (for example, entering Oracel instead of Oracle). Also, synonyms can provide more relevant results; for example, cellular phones for cell phones or wireless phones. Additional uses for alternate keywords are for product code names and abbreviations.

- **Search - Source Groups Page**

  Set options to allow users to limit their searches. For example, searches can be limited to document attributes, such as title or author. Searches can also be limited to source groups. Source groups are logical entities exposed to end users. When entering a search, they can select one or more source groups from which to search. Each source group consists of one or more sources. If no source group is selected, then all documents are searched.

Global Settings Tab

The **Global Settings** tab includes links to configure settings for your Oracle SES environment.



This page configures various settings for your Oracle SES environment. This section describes some of the global configuration pages.

- **Crawler Configuration Page**

  This page configures global crawler settings, such as crawling depth, language, and maximum document size.

  After a source has been created, you can define crawling parameters, such as URL boundary rules and crawling depth, for that source by editing that source on the **Home - Sources** page.

  See Also: "Overview of Crawler Settings" on page 3-2

- **Query Configuration Page**

  This page includes the following options:

  - Maximum number of results returned to users.
  - Display URL - For example, with table sources, when gathering information from a database Web application, Oracle SES lets you specify a URL to display the retrieved data on a browser.
  - Spell checking - This suggests corrections to end users based on data available from an English language dictionary and crawled data.
  - Statistics collection - The logging of search statistics reduces search performance, so consider disabling this during regular operation.
  - URL submission - This lets end users submit URLs to be crawled and indexed. You can examine submitted URLs before they are indexed by the crawler.
  - Federated search - This allows multiple indexes to perform a single search.
  - Security filter configuration - Oracle SES supports identity-based security filters using users and groups from an identity management system.

- **Identity Management Setup Page**

  This page lets you set up connections between Oracle Secure Enterprise Search and any identity management system to validate and authenticate users. This is necessary for secure searches. Oracle SES uses an *identity plug-in* as an interface to

an identity management system. Oracle SES provides a registered identity plug-in for Oracle Internet Directory.

See Also:

- Oracle SES admin tutorial for help with common administrator tasks:

  http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm

- Oracle SES administration tool context sensitive online help
- Oracle SES home page for updated information on known issues, as well as code samples and best practices:

  http://www.oracle.com/technology/products/oses/index.html

3

Understanding Crawling and Searching

This chapter contains the following topics:

- Overview of the Oracle Secure Enterprise Search Crawler
- Overview of Crawler Settings
- Overview of Attributes
- Understanding the Crawling Process
- Monitoring the Crawling Process
- Overview of Searching in Oracle Secure Enterprise Search

See Also:

- "Tuning Crawl Performance" on page 5-9 and "Tuning Search Performance" on page 5-13
- The Oracle Secure Enterprise Search tutorials at `http://www.oracle.com/technology/products/oses/index.html`

Overview of the Oracle Secure Enterprise Search Crawler

The Oracle Secure Enterprise Search (SES) crawler is a Java process activated by a set schedule. When activated, the crawler spawns processor threads that fetch documents from **sources**. These documents are cached in the local file system. When the cache is full, the crawler indexes the cached files. This index is used for searching.

In the administration tool, you can create schedules with one or more sources attached to them. Schedules define the frequency at which the Oracle SES index is kept up to date with existing information in the associated sources.

Crawler URL Queue

In the process of crawling, the crawler maintains a list of URLs of the documents that are discovered and will be fetched and indexed in an internal URL queue. The queue is persistently stored, so that crawls can be resumed after the Oracle SES instance is restarted.

Understanding Access URLs and Display URLs

A display URL is a URL string used for search result display. This is the URL used when users click the search result link. An access URL is a URL string used by the crawler for crawling and indexing. An access URL is optional. If it does not exist, then

the crawler uses the display URL for crawling and indexing. If it does exist, then it is used by the crawler instead of the display URL for crawling. For regular Web crawling, there are only display URLs available. But in some situations, the crawler needs an access URL for crawling the internal site while keeping a display URL for the external use. For every internal URL, there is an external mirrored one.

For example, for file sources, by defining display URLs, end users can access the original document with the HTTP or HTTPS protocols. These provide the appropriate authentication and personalization and result in better user experience.

Display URLs can be provided using the URL Rewriter API. Or, they can be generated by specifying the mapping between the prefix of the original file URL and the prefix of the display URL. Oracle SES replaces the prefix of the file URL with the prefix of the display URL. For example, if the file URL is file://localhost/home/operation/doc/file.doc and the display URL is https://webhost/client/doc/file.doc, then specify the file URL prefix to file://localhost/home/operation and the display URL prefix to https://webhost/client.

Using Crawler Plug-ins

In addition to the default source types Oracle SES provides (such as Web, file, OracleAS Portal, and so on), you can also crawl proprietary sources, such as Lotus Notes or Documentum. This is accomplished by implementing a crawler plug-in as a Java class. The plug-in collects document URLs and associated metadata (including access privilege) and contents from the proprietary source and returns the information to the Oracle SES crawler. The crawler starts processing each document as it is collected.

See Also: "Crawler Plug-in API" on page 6-25

Overview of Crawler Settings

You can alter the crawler's operating parameters, such as the crawler timeout threshold and the default character set, on the **Global Settings - Crawler Configuration** page in the administration tool.

This section describes crawler settings, as well as other mechanisms to control the scope of Web crawling:

- Crawling Mode
- URL Boundary Rules
- Crawling Depth
- Robots Exclusion
- Index Dynamic Pages
- URL Rewriter API
- Title Fallback

See Also: "Tuning Crawl Performance" on page 5-9 for more detailed information on these settings and other issues affecting crawl performance

Crawling Mode

For initial planning purposes, you might want the crawler to collect URLs without indexing. After crawling is finished, examine the document URLs and status, remove unwanted documents, and start indexing. The crawling mode is set on the **Home - Schedules - Edit Schedules** page.

See Also: Appendix A, "URL Crawler Status Codes"

Note: If you are using a custom crawler created with the Crawler Plug-in API, then the crawling mode set here will not apply. The implemented plug-in controls the crawling mode.

These are the crawling mode options:

- **Automatically Accept All URLs for Indexing**: This crawls and indexes all URLs in the source. For Web sources, it also extracts and indexes any links found in those URLs. If the URL has been crawled before, then it will be reindexed only if it has changed.
- **Examine URLs Before Indexing**: This crawls but does not index any URLs in the source. It also crawls any links found in those URLs.
- **Index Only**: This crawls and indexes all URLs in the source. It does not extract any links from those URLs. In general, select this option for a source that has been crawled previously under "Examine URLs Before Indexing".

URL Boundary Rules

URL boundary rules limit the crawling space. When boundary rules are added, the crawler is restricted to URLs that match the indicated rules. The order in which rules are specified has no impact, but exclusion rules always override inclusion rules.

This is set on the **Home - Sources - Boundary Rules** page.

Inclusion Rules

Specify an inclusion rule that a URL contain, start with, or end with a term. Use an asterisk (*) to represents a wildcard. For example, www.*.example.com. Simple inclusion rules are case-insensitive. For case-sensitivity, use regular expression rules.

An inclusion rule ending with example.com limits the search to URLs ending with the string example.com. Anything ending with example.com is crawled, but http://www.example.com.tw is not crawled.

If the URL Submission functionality is enabled on the **Global Settings - Query Configuration** page, then URLs that are submitted by end users are added to the inclusion rules list. You can delete URLs that you do not want to index.

Oracle SES supports the regular expression syntax used in Java JDK 1.4.2 Pattern class (java.util.regex.Pattern). Regular expression rules use special characters. The following is a summary of some basic regular expression constructs.

- Use a caret (^) to denote the beginning of a URL and a dollar sign ($) to denote the end of a URL.
- Use a period (.) to match any one character.
- Use a question mark (?) to match zero or one occurrence of the character that it follows.

- Use an asterisk (*) to match zero or more occurrences of the pattern that it follows. An asterisk can be used in the starts with, ends with, and contains rule.
- Use a backslash (\) to escape any special characters, such as periods (\.), question marks (\?), or asterisks (*).

> **See Also:** `http://java.sun.com` for a complete description on Sun Microsystems Java documentation

Exclusion Rules

You can specify an exclusion rule that a URL contains, starts with or ends with a term.

An exclusion of uk.example.com prevents the crawling of Example hosts in the United Kingdom.

Default Exclusion Rules

The crawler contains a default exclusion rule to exclude non-textual files. The following file extensions are included in the default exclusion rule.

- Image: jpg, gif, tif, bmp, png
- Audio: wav, mp3, wma
- Video: avi, mpg, mpeg, wmv
- Binary: bin, exe, so, dll, iso, jar, war, ear, tar, wmv, scm, cab, dmp

Example Using Regular Expression

The following example uses several regular expression constructs which are not described earlier, including range quantifiers, non-grouping parentheses, and mode switches. For a complete description, see the Sun Microsystems Java documentation.

Suppose you want to crawl only HTTPS URLs within the example.com and examplecorp.com domains. Also, you want to exclude files ending in .doc and .ppt.

- Inclusion: URL regular expression ^https://.*\.example(?:corp){0,1}\.com
- Exclusion: URL regular expression (?i:\.doc|\.ppt)$

Crawling Depth

Crawling depth is the maximum number of nested links the crawler will follow. (A Web document could contain links to other Web documents, which could contain more links.)

This is set on the **Home - Sources - Crawling Parameters** page.

Robots Exclusion

You can control which parts of your sites can be visited by robots. If robots exclusion is enabled (default), then the Web crawler traverses the pages based on the access policy specified in the Web server `robots.txt` file. The crawler also respects the page-level robot exclusion specified in HTML metatags.

For example, when a robot visits `http://www.example.com/`, it checks for `http://www.example.com/robots.txt`. If it finds it, then the crawler checks to see if it is allowed to retrieve the document. If you own the Web sites, then you can disable robots exclusions. However, when crawling other Web sites, always comply with `robots.txt` by enabling robots exclusion.

This is set on the **Home - Sources - Crawling Parameters** page.

Index Dynamic Pages

By default, Oracle SES will process dynamic pages. Dynamic pages are generally served from a database application and have a URL that contains a question mark (?). Oracle SES identifies URLs with question marks as dynamic pages.

Some dynamic pages appear as multiple search results for the same page, and you might not want them all indexed. Other dynamic pages are each different and need to be indexed. You must distinguish between these two kinds of dynamic pages. In general, dynamic pages that only change in menu expansion without affecting its contents should not be indexed. Consider the following three URLs:

```
http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html

http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_
convention.html?nsdnv=14z1

http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_
convention.html?nsdnv=14
```

The question mark ('?') in the URL indicates that the rest of the strings are input parameters. The duplicate results are essentially the same page with different side menu expansion. Ideally, the search should yield only one result:

```
http://itweb.oraclecorp.com/aboutit/network/npe/standards/naming_convention.html
```

Note: The crawler cannot crawl and index dynamic Web pages written in Javascript.

This is set on the **Home - Sources - Crawling Parameters** page.

URL Rewriter API

The URL Rewriter is a user-supplied Java module for implementing the Oracle SES `UrlRewriter` interface. The crawler uses it to filter or rewrite extracted URL links before they are put into the URL queue. The API enables ultimate control over which links extracted from a Web page are allowed and which ones should be discarded.

URL filtering removes unwanted links, and URL rewriting transforms the URL link. This transformation is necessary when access URLs are used and alternate display URLs need to be presented to the user in the search results.

This is set on the **Home - Sources - Crawling Parameters** page.

See Also:

- "URL Rewriter API" on page 6-28
- *Oracle Secure Enterprise Search Java API Reference*

Title Fallback

You can override a default document title with a meaningful title if the default title is irrelevant. For example, suppose that the result list shows numerous documents with the title "Daily Memo". The documents had been created with the same template file, but the document properties had not been changed. Overriding this title in Oracle SES can help users better understand their search results.

Title fallback can be used for any source type. Oracle SES uses different logic for each document type to determine which fallback title to use. For example, for HTML documents, Oracle SES looks for the first heading, such as <h1>. For Microsoft Word documents, Oracle SES looks for text with the largest font.

If the default title was collected in the initial crawl, then the fallback title will only be used after the document is reindexed during a recrawl. This means if there is no change to the document, then you must force the change by setting the recrawl policy to **Process All Documents** on the **Home - Schedules - Edit Schedule** page.

This feature is not currently supported in the Oracle SES administration tool. Configure title fallback in the crawler configuration file: `$ORACLE_HOME/search/data/config/crawler.dat`.

Note: The replaced title cannot be searched with the title attribute on the **Advanced Search** page.

See Also: "Replacing Default Document Titles" on page 3-10

Overview of Attributes

Each source has its own set of document attributes. Document attributes, like metadata, describe the properties of a document. The crawler retrieves values and maps them to one of the search attributes. This mapping lets users search documents based on their attributes. Document attributes in different sources can be mapped to the same search attribute. Therefore, users can search documents from multiple sources based on the same search attribute.

Document attribute information is obtained differently depending on the source type. For example, with Web sources, document attributes are extracted from HTML META tags. With table sources, any column in the source table can be chosen as a document attribute. With user-defined sources, document attributes and values can be returned by the crawler plug-in module.

Document attributes can be used for many things, including document management, access control, or version control. Different sources can have different attribute names which are used for the same idea; for example, "version" and "revision". It can also have the same attribute name for different ideas; for example, "language" as in natural language in one source but as programming language in another.

Oracle SES has several default search attributes. They can be incorporated in search applications for a more detailed search and richer presentation.

Search attributes are defined in the following ways:

- System-defined search attributes, such as title, author, description, subject, and mimetype
- Search attributes created by the Oracle SES administrator
- Search attributes created by the crawler. (During crawling, the crawler plug-in maps the document attribute to a search attribute with the same name and data type. If not found, then the crawler creates a new search attribute with the same name and type as the document attribute defined in the crawler plug-in.)

The list of values (LOV) for a search attribute can help you specify a search. Global search attributes can be specified on the **Global Settings - Search Attributes** page. For user-defined sources where LOV information is supplied through a crawler plug-in, the crawler registers the LOV definition. Use the administration tool or the crawler

plug-in to specify attribute LOVs, attribute value, attribute value display name, and its translation.

> **Note:** When multiple sources define the LOV for a common attribute, such as title, the user sees all the possible values for the attribute. When the user restricts search within a particular source group, only LOVs provided by the corresponding sources in the source group will be shown.

Understanding the Crawling Process

The first time the crawler runs, it must fetch data (Web pages, table rows, files, and so on) based on the source. It then adds the document to the Oracle SES index.

The Initial Crawl

This section describes a Web source crawling process for a schedule. It is broken into two phases:

1. Queuing and Caching Documents
2. Indexing Documents

Queuing and Caching Documents

The steps in the crawling cycle are the following:

1. Oracle spawns the crawler according to the schedule you specify with the administration tool. When crawling is initiated for the first time, the URL queue is populated with the seed URLs.
2. The crawler initiates multiple crawling threads.
3. The crawler thread removes the next URL in the queue.
4. The crawler thread fetches the document from the Web. The document is usually an HTML file containing text and hypertext links.
5. The crawler thread scans the HTML file for hypertext links and inserts new links into the URL queue. Duplicate links already in the document table are discarded.
6. The crawler caches the HTML file in the local file system.
7. The crawler registers URL in the URL table.
8. The crawler thread starts over by repeating Step 3.

Fetching a document, as described in Step 4, can be time-consuming because of network traffic or slow Web sites. For maximum throughput, multiple threads fetch pages at any given time.

Indexing Documents

When the file system cache is full (default maximum size is 250 MB), the indexing process begins. At this point, the document content and any searchable attributes are pushed into the index. After indexing of the document in the batch is completed, the crawler switches back to the queuing and caching mode.

Maintenance Crawls

After the initial crawl, a URL page is only crawled and indexed if it has changed since the last crawl. The crawler determines if it has changed with the HTTP If-Modified-Since header field or with the checksum of the page. URLs that no longer exist are marked and removed from the index.

To update changed documents, the crawler uses an internal checksum to compare new Web pages with cached Web pages. Changed Web pages are cached and marked for reindexing.

The steps involved in data synchronization are the following:

1. Oracle spawns the crawler according to the schedule you specify with the administration tool. The URL queue is populated with the seed URLs of the source assigned to the schedule.
2. The crawler initiates multiple crawling threads.
3. Each crawler thread removes the next URL in the queue.
4. Each crawler thread fetches the document from the Web. The page is usually an HTML file containing text and hypertext links. When the document is not in HTML format, the crawler tries to convert the document into HTML before caching.
5. Each crawler thread calculates a checksum for the newly retrieved page and compares it with the checksum of the cached page. If the checksum is the same, then the page is discarded and the crawler goes to Step 3. Otherwise, the crawler moves to the next step.
6. Each crawler thread scans the document for hypertext links and inserts new links into the URL queue. Links that are already in the document table are discarded. (Oracle SES does not follow links from filtered binary documents.)
7. The crawler marks the URL as "accepted". The URL will be crawled in future maintenance crawls.
8. The crawler registers the URL in the document table.
9. If the file system cache is full or if the URL queue is empty, then Web page caching stops and indexing begins. Otherwise, the crawler thread starts over at Step 3.

Monitoring the Crawling Process

Monitor the crawling process in the administration tool by using a combination of the following:

- Check the crawl progress and crawl status on the **Home - Schedules** page. (Click **Refresh Status**.)
- Monitor your crawler statistics on the **Home - Schedules - Crawler Progress Summary** page and the **Home - Statistics** page.
- Monitor the log file for the current schedule.

See Also: "Tuning Crawl Performance" on page 5-9

Crawler Statistics

The following crawler statistics are shown on the **Home - Schedules - Crawler Progress Summary** page. Some of these statistics are also shown in the log file, under "Crawling results".

- Documents to Fetch: Number of URLs in the queue waiting to be crawled. The log file uses the term "Documents to Process".
- Documents Fetched: Number of documents retrieved by the crawler.
- Document Fetch Failures: Number of documents whose contents cannot be retrieved by the crawler. This could be due to an inability to connect to the Web site, slow server response time causing timeouts, or authorization requirements. Problems encountered after successfully fetching the document are not considered here; for example, documents that are too big or documents ignored due to duplicates.
- Documents Rejected: Number of URL links encountered but not considered for crawling. The rejection could be due to boundary rules, the robots exclusion rule, the mime type inclusion rule, the crawling depth limit, or the URL rewriter discard directive.
- Documents Discovered: All documents discovered during crawling. This is roughly equal to (documents to fetch) + (documents fetched) + (document fetch failures) + (documents rejected).
- Documents Indexed: Number of documents that have been indexed or are pending indexing.
- Documents non-indexable: Number of documents that cannot be indexed; for example, a file source directory or a document with robots NOINDEX metatag.
- Document Conversion Failures: Number of document filtering errors. This is counted whenever a document cannot be converted to HTML format.

Crawler Log File

The log file records all crawler activity, warnings, and error messages for a particular schedule. It includes messages logged at startup, runtime, and shutdown. Logging everything can create very large log files when crawling a large number of documents. However, in certain situations, it can be beneficial to configure the crawler to print detailed activity to each schedule log file.

A new log file is created when you restart the crawler. The crawler maintains the past seven versions of its log file, but only the most recent log file is shown in the administration tool. You can view the other log files in the file system. The location of the crawler log file can be found on the **Home - Schedules - Crawler Progress Summary** page.

The naming convention of the log file name is `ids.MMDDhhmm.log`, where `ids` is a system-generated ID that uniquely identifies the source, `MM` is the month, `DD` is the date, `hh` is the launching hour in 24-hour format, and `mm` is the minutes.

For example, if a schedule for a source identified as i3ds23 is launched at 10 pm, July 8th, then the log file name is `i3ds23.07082200.log`. Each successive schedule launching will have a unique log file name. If the total number of log files for a source reaches seven, then the oldest log file is deleted.

Each logging message in the log file is one line, containing the following six tab delimited columns, in order:

1. Timestamp
2. Message level
3. Crawler thread name
4. Component name. It is in general the name of the executing Java class.

5. Module name. It can be internal Java class method name
6. Message

> **Note:** Debug information is found in the OC4J log file: `$ORACLE_HOME/oc4j/j2ee/OC4J_SEARCH/log/oc4j.log`

Crawler Configuration File

The crawler configuration file is `$ORACLE_HOME/search/data/config/crawler.dat`.

> **Note:** The `crawler.dat` file is not backed up with Oracle SES backup and recovery. If you edit this file, make sure to back it up manually.

Setting the Logging Level Specify the crawler logging level with the parameter `Doracle.search.logLevel`. The defined levels are `DEBUG(2)`, `INFO(4)`, `WARN(6)`, `ERROR(8)`, `FATAL(10)`. The default value is 4, which means that messages of level 4 and higher will be logged. DEBUG (level=2) messages are not logged by default.

For example, the following "info" message is logged at 23:10:39330 and has logging level 4. It is from thread name `crawler_2`, and the message is `Processing file://localhost/net/stawg02/`. The component and module names are not specified.

```
23:10:39:330 4    crawler_2      Processing file://localhost/net/stawg02/
```

The crawler uses a set of codes to indicate the crawling result of the crawled URL. Besides the standard HTTP status codes, it uses its own codes for non-HTTP related situations.

> **See Also:** Appendix A, "URL Crawler Status Codes"

Replacing Default Document Titles Override a default document title with a meaningful title by adding the keyword `BAD_TITLE` to the `crawler.dat` file. For example:

```
BAD_TITLE Daily Memo
```

Where *`Daily Memo`* is the title string that should be overridden. The title string is case-insensitive and can use multibyte characters.

Multiple bad titles can be specified, each one on a separate line.

> **See Also:** "Title Fallback" on page 3-5 for more information on this feature

Overview of Searching in Oracle Secure Enterprise Search

To get to the end user search page from any page in the administration tool, click the **Search** link in the top right corner. This brings up the Basic Search page in a new window, with a text box to enter a search string.

This section contains the following topics:

- Basic Search
- Advanced Search
- Browse Source Groups
- Submit URL

See Also: "Tuning Search Performance" on page 5-13

Basic Search

The search string can consist of one or more words. Clicking the search button returns all matches for that search string. The results can include the following links:

Cached: The cached HTML version of the document.

Links: Pages that link to and from this document.

Source Group: This link leads to Browse Source Groups.

Any links on top of the search text box are source groups. Clicking a source group restricts the search to that group.

The following table describes rules that apply to the search string. Text in square brackets represents characters entered into the search.

Table 3–1 Search String Rules

Rule	Description
Single word search	Entering one word finds documents that contain that word. For example, searching for [Oracle] finds all documents that contain the word Oracle anywhere in that document.
Compulsory inclusion [+]	Attaching a [+] in front of a word requires that the word be found in all matching documents. For example, searching for [Oracle +Applications] only finds documents that contain the words Oracle and Applications. Note: in a multiple word search, you can attach a [+] in front of every token including the very first token. A token is a phrase enclosed in double-quotes ("). It can be a single word or a phrase, but there should be no space between the [+] and the token.
Compulsory exclusion [-]	Attaching a [-] in front of a word requires that the word not be found in all matching documents. For example, searching for [Oracle -Applications] only finds documents that contain the word Oracle and *not* the word Applications. Note: in a multiple word search, you can attach a [-] in front of every token except the very first token. A token is a phrase enclosed in double-quotes ("). It can be a single word or a phrase, but there should be no space between the [-] and the token.
Phrase matching ["..."]	Putting quotes around a set of words only finds documents that contain that precise phrase. For example, searching for ["Oracle Applications"] only finds documents that contain the string Oracle Applications.

Table 3–1 (Cont.) Search String Rules

Rule	Description
Wildcard matching [*]	Attaching a [*] to the right side of a word returns left side partial matches. For example, searching for the string [Ora*] finds documents that contain all words beginning with Ora, such as Oracle and Orator. You can also insert an asterisk in the middle of a word. For example, searching for the string [A*e] finds documents that contain words such as Apple or Ape. Wildcard matching cannot be used with Chinese or Japanese native characters.
Site search	Attaching [site:host] after the search term limits results to that particular site. For example, "documentation site:www.oracle.com". Oracle SES supports exact host matching (that is, site:*.oracle.com does not work) and one "site:" for each search.
File type filtering	Attaching [filetype:filetype] after the search term limits results to that particular file type. For example, "documentation filetype:pdf", returns PDF format documents for the term documentation. A search can have only one filetype shortcut. The following file types are supported (with their corresponding "string"): filetype string: mimetype ps: application/postscript ppt: application/vnd.ms-powerpoint, application/x-mspowerpoint doc: application/msword xls: application/vnd.ms-excel, application/x-msexcel, application/ms-excel txt: text/plain html: text/html htm: text/html pdf: application/pdf xml: text/xml rtf: application/rtf

Advanced Search

The Advanced Search page lets you refine searches in the following ways:

- Narrowing Searches by Search Attributes
- Limiting Searches to Certain Source
- Limiting Searches to Documents Written in a Specific Language

Narrowing Searches by Search Attributes

With the Advanced Search page, you can require that documents matching your search have specific attributes values. To specify a search attribute value, use the list boxes to select a search attribute. Enter the search attribute value in the text box immediately to the right of the list box. Date format must be entered as MM/DD/YYYY format.

Limiting Searches to Certain Source

If one or more source groups are defined, then corresponding check boxes appear when you select specific categories. You can limit your search to source groups by selecting those check boxes. If no source group is selected, then all documents are searched. If you select **All**, (that is, all source groups present), then the documents not in the selected groups (in the default group) will not be searched.

A source group represents a collection of documents. They are created by the Oracle SES administrator.

Limiting Searches to Documents Written in a Specific Language

Oracle SES can search documents in different languages. Specifying a language restricts searches to documents that are written in that language. Use the language list box to specify a language.

Browse Source Groups

Source groups are groups of sources that can be searched together. A source group consists of one or more sources, and a source can be assigned to multiple source groups. Source groups are defined on the **Search - Source Groups** page. Groups, or folders, are only generated for Web, e-mail, and OracleAS Portal source types.

On **Search** page, users can browse source groups that the administrator created. Click a source group name to see the subgroups under it, or drill down further into the hierarchy by clicking a subgroup name.

To view all the documents under a particular group, click the number next to the source group name. You can also perform a restricted search within the source group from this page.

The source hierarchy lets end users limit search results based on document source type. The hierarchy is generated automatically during crawl time.

Submit URL

The URL submission feature lets users submit URLs to be crawled and indexed. These URLs are added to the seed URL list for a particular source and included in the crawler search space.

If you allow URL submission (on the **Global Settings - Query Configuration** page), then you must select the Web source to which submitted URLs will be added.

Note: This feature is disabled on the **Search** page if no sources have been created.

4

Security in Oracle Secure Enterprise Search

This chapter describes the architecture and configuration for Oracle Secure Enterprise Search (SES) security model.

This chapter contains the following sections:

About Oracle Secure Enterprise Search Security

This section describes the Oracle SES security model. It contains the following sections:

Oracle Secure Enterprise Search Security Model

Oracle SES provides access to a variety of content repositories through a single gateway. Each one of these external repositories has its own security model that determines whether a particular user can access a particular document. All the aspects of security in Oracle SES must be carefully considered to respect the security of documents coming from multiple data repositories.

Note: Connecting to the Oracle SES server using SQL*Plus, except as documented in the guide, is not supported. As an additional security measure, Oracle SES is configured to reject connection requests using SQL*Plus from remote hosts. The only protocols supported for connection to Oracle SES from remote hosts are HTTP, HTTPS, and AJP13. Changing the Oracle SES server directly using SQL and modifying initialization parameter files is not supported. User management, including password changes, should only be done using the Oracle SES administration tool.

How Oracle Secure Enterprise Search Leverages Security Services

Oracle SES uses the following security services in its security model:

- Secure socket layers (SSL): This is the industry standard protocol for managing the security of message transmission on the Internet. This is used for securing RMI connections, HTTPS crawling, and secure JDBC.
- Oracle Internet Directory LDAP APIs: These provide application developers with user authentication and authorization services to integrate them into their environments.

Temporary Passwords

For added security, a temporary password feature is provided. When creating table sources, e-mail, OracleAS Portal, or Web sources, login credentials for use by the crawler can be entered. (For Web sources, authentication can be performed with HTTP authentication, HTML forms, and OracleAS Single Sign-On.) These passwords can be removed from the Oracle SES repository after the schedule they are in completes. To use the temporary password feature, click the option to **Delete Passwords After Crawl** when creating or editing the source. This option is not available if self service for Web sources is enabled.

If a source has the **Delete Passwords after Crawl** option enabled, then the administrator will be prompted for all required passwords whenever the schedule for that source is launched. The supplied passwords will be removed immediately after the corresponding schedule completes. Because the administrator will be prompted for the passwords when launching the crawler, schedules containing sources having the **Delete Passwords after Crawl** option enabled must be launched manually.

Authorization and Authentication

Note: With release 10*g* Release 1 (10.1.7) Beta, Oracle SES can use an *identity plug-in* to obtain user and group information directly from any identity management system. (Oracle SES no longer requires access control lists in Oracle Internet Directory for secure search.) An identity plug-in is Java code that sits between Oracle SES and an identity management system, allowing Oracle SES to read user and group information.

Security filter configuration for the identity plug-in is done on the **Global Settings - Query Configuration** page.

Oracle SES security is implemented at the following levels:

- User authentication

  This is the identification of a user through an identity management system. Oracle SES lets you register an identity plug-in as an interface to *any* identity management system. (Oracle SES provides a registered identity plug-in for Oracle Internet Directory.) The plug-in that you activate is responsible for all authentication and validation activity in Oracle SES. This is done on the **Global Settings - Identity Management Setup** page.

- User authorization

  This determines whether a user can access information about a particular item in the results list. It is implemented in two layers.

The first layer utilizes access control lists (ACLs). An ACL lists which users or groups of users have access to the document. The ACL can be assigned by the administrator to an entire source through the administration tool (*source-level ACLs*), or it can be provided by the source itself for each document (*document-level ACLs*).

The second layer uses a Java class to dynamically filter documents at search time (query-time filtering).

Oracle SES can make use of the following types of ACL policies:

– Source-level ACLs: These are defined on the **Home - Sources - Authorization** page. An individual source can be protected by a single ACL, which governs access to every document in that source.

– Document-level ACLs: Oracle SES provides mapped security to repositories by retrieving the ACL for each document at the time of crawling and indexing. At crawl time, the ACL for each document is passed to the crawler along with the document content, and the ACL is stored in the index. Currently Oracle SES supports document-level ACLs for user-defined sources and OracleAS Portal sources. (The ACL policy is **Documents Controlled by the Source**.) With user-defined sources, ACLs are returned by the crawler plug-in implemented by the user. With OracleAS Portal sources, ACLs are returned by the OracleAS Portal server. At search time, Oracle SES does not need any connection with the repository to validate access privileges.

Note: For both source-level ACLs and document-level ACLs, all users and roles defined in the ACLs must exist in the identity plug-in.

The following table compares the document-level user authorization methods in Oracle SES.

Table 4–1 User Authorization Methods in Oracle Secure Enterprise Search

Method	How Authorization is Determined	Advantages	Disadvantages
ACLs	The ACL is supplied by a crawler plug-in or an OracleAS Portal server.	Faster secure search performance. No additional programming is required for ACL-based OracleAS Portal security. (If implementing a crawler plug-in, then some additional work is necessary to supply ACLs.)	ACLs are static: they are updated only when crawling the source repository or when the administrator changes Oracle SES ACLs in the administration tool
Query-time Authorization	`QueryTimeFilter` Java class.	Dynamic authorization. Reflects real-time user access privilege on documents.	There is performance overhead in cases when the search is not selective, returning large number of rows before query-time filtering. Extra work is required to implement a `QueryTimeFilter`.

See Also:

- "Admin-based Authorization" on page 4-7 for more information about using ACLs
- "Query-time Authorization" on page 4-8 for more information on using a Java filter class
- "Crawler Plug-in API" on page 6-25

Restrictions on Changing the ACL Policy

On the **Home - Sources - Authorization** page, you can set and change the ACL policy. The following ACL policy options are available:

- **No ACL**: With this setting, all documents are considered searchable and visible
- **Oracle Secure Enterprise Search ACL**: With this setting (also known as *source-level ACLs*), you can protect the entire source with one ACL. The same ACL protects every document in that source.
- **ACLs Controlled by the Source**: This setting (also known as *document-level ACLs*) is available only for OracleAS Portal sources and user-defined sources. This preserves authorizations specified in OracleAS Portal. For user-defined sources, crawler plug-ins (or *connectors*) can supply ACL information together with documents for indexing, which provides finer control document protection. (That is, each document in the source can have different access privileges.)

The following restrictions apply to changing the ACL policy:

- If the schedule associated with that source is currently being crawled (that is, the schedule status is **Launching**, **Executing**, or **Stopping**), then all ACL options are grayed out, and you cannot change the ACL policy.
- If the schedule associated with the source is not currently being crawled, and if the source has never been crawled, then all ACL policy changes are allowed.
- If the schedule associated with the source is not currently being crawled, but the source *has* been crawled in the past, then the only change allow is between **No ACL** and **Oracle Secure Enterprise Search ACL** (in either direction). This is visible in the administration tool as follows:
  - If the ACL option selected before the crawl started was **ACLs Controlled by the Source**, then all options are grayed out.
  - If the ACL option selected before the crawl started was **No ACL** or **Oracle Secure Enterprise Search ACL**, then the **ACLs Controlled by the Source** option is grayed out.
- OracleAS Portal sources are subject to the same restrictions as other sources. That is, no changes are allowed while being crawled, and only changes between **No ACL** and **Oracle Secure Enterprise Search ACL** are allowed after crawling completes. However, the ACL policy for an OracleAS Portal source can also change if it is inheriting the ACL policy from its OracleAS Portal server parent; for example, when the OracleAS Portal server's ACL policy is modified or when the OracleAS Portal source is changed from specifying the ACL policy locally to inheriting it from the server. Therefore, changes on an OracleAS Portal server are restricted so that no disallowed changes can occur on any children that inherit the ACL policy. If any child inheriting the ACL policy is being crawled, then no changes are allowed on the OracleAS Portal server. If any child inheriting the ACL policy has already been crawled, then the only changes allowed are between **No ACL** and **Oracle Secure Enterprise Search ACL**. (If the OracleAS Portal server

policy is **ACLs Controlled by the Source**, then no changes are allowed). Similarly, the OracleAS Portal source cannot be set to inherit its ACL policy from the OracleAS Portal server if the associated change in ACL policy would be disallowed.

Note: There is a difference between *a source that is being crawled* and *a source whose associated schedule is being crawled*. Oracle SES restricts all ACL policy changes for a source when the schedule associated with that source is being crawled. A source might not be crawling, but the schedule associated with it could be crawling if another source in the same schedule is being crawled.

Restrictions on Changing the Oracle Internet Directory Identity Plug-in

The information Oracle SES saves from Oracle Internet Directory (that is, the correspondence between names and GUIDs) may not be valid on different Oracle Internet Directory servers. If you keep the same Oracle Internet Directory server (for example, to change port numbers or to switch to SSL), or if you use a new directory server that has identical user information, then you can deactivate and re-activate the Oracle Internet Directory plug-in anytime without restriction. This section describes steps you must perform if you change Oracle Internet Directory servers with user information that is not identical.

If you have sources using the ACL policy **Oracle Secure Enterprise Search ACL** and you decide to change your active Oracle Internet Directory plug-in to use a different Oracle Internet Directory server, then you must clear the ACL data before deactivating the original Oracle Internet Directory plug-in. If the ACL data is not cleared, then the ACL policy configured for that source while connected to the old Oracle Internet Directory server will not be correctly enforced after connecting to the new Oracle Internet Directory server.

The existing ACL data can be cleared using either of the following two ways:

- Before deactivating the Oracle Internet Directory plug-in, for each source using the ACL policy **Oracle Secure Enterprise Search ACL**, switch the ACL policy to **No ACL**. After connecting to the new Oracle Internet Directory server, restore the ACL policy to **Oracle Secure Enterprise Search ACL** and add the ACLs again. NOTE: This will temporarily make the source public. If this is unacceptable, then use the next option.
- Before deactivating the Oracle Internet Directory plug-in, delete each source that uses the ACL policy **Oracle Secure Enterprise Search ACL**. After connecting to the new Oracle Internet Directory server, add the sources back and configure them again. The documents are never made public; but this may involve more work than the previous option.

If you have sources using the ACL policy **ACLs Controlled by the Source** and you decide to change your active Oracle Internet Directory plug-in to use a different Oracle Internet Directory server, then after activating the new Oracle Internet Directory server, each source that uses this ACL policy must be re-crawled with the **Process All Documents** option. This forces the reloading and indexing of all of ACL information for such sources with respect to the new Oracle Internet Directory server. Set the **Process All Documents** option on the **Home - Schedules - Edit Schedule** page.

Note: if the ACL data is not cleared before switching Oracle Internet Directory servers, then you will see a message that some users and groups could not be found by the identity plug-in. Those users and groups are still displayed on the **Home - Sources -Authorization** page. They can be manually deleted.

Authentication Methods

Oracle SES authentication for search users has two aspects: a user repository at the back-end and an authentication user interface at the front-end. The front-end interface collects user credentials, which are then validated against the back-end repository.

In addition to authentication of search users, Oracle SES must also authenticate the crawler when accessing external data repositories. Administrators supply credentials to crawl private content through the following authentication methods:

- HTTP authentication (both basic and digest authentication)
- HTML forms
- OracleAS Single Sign-On

It is the administrator's responsibility to check the authorization policy to make sure that crawled documents are properly protected.

Oracle Secure Enterprise Search User Repository

Oracle SES has two types of users:

1. **Administrative User**: The administrative user is called `EQSYS`. This user is natively defined in Oracle SES. Only this user can use the administration tool.
2. **Search Users**: Oracle SES lets you register an identity plug-in as an interface to any identity management system. (By default, Oracle SES provides a registered identity plug-in for Oracle Internet Directory.) The plug-in that you activate is responsible for all authentication and validation activity in Oracle SES. Use the **Global Settings - Identity Management Setup** page in the administration tool to associate Oracle SES with an identity management system.

Note: Oracle Internet Directory is Oracle's native LDAP v3-compliant directory service. It is part of the Oracle Identity Management infrastructure. Oracle Internet Directory should be version 9.0.4 or 10.1.2 (with the latest patch release applied) for connection with Oracle SES. Oracle Internet Directory is not a part of Oracle SES, and therefore Oracle SES can be linked to any existing or new Oracle Internet Directory.

Oracle Secure Enterprise Search Authentication Interface

For the administrative user `EQSYS`, there is a form login screen in the Oracle SES administration tool. This is the only way an administrative user can log in to Oracle SES.

For search users, there are three possible front-end authentication interfaces:

- HTML form login page. Oracle SES provides an authentication page, and it authenticates against the identity plug-in.

- Web Services API. The `login` and `logout` Web Services operations authenticate against the identity plug-in.
- Single sign-on login screen. This can be made available by front-ending Oracle SES with OracleAS Single Sign-On and Oracle HTTP Server. These are available as part of the Oracle Identity Management infrastructure in OracleAS.

Note:

- Only form login *or* single sign-on login can be used for search users at any point in time. Using single sign-on with the Web Services authentication interface is not supported.
- Oracle strongly recommends that you SSL-protect the channel between the Oracle HTTP Server and the Oracle SES OC4J instance for secure content.
- Oracle SES does not include Oracle Internet Directory: It should be installed separately.

Enabling Secure Search

Much of the information within an organization is publicly accessible. However, there are other sources that are protected. For example, while a user can search within their own e-mail folders, they should not be able to search anyone else's e-mail. A secure search returns only search results that the user is allowed to view based on access privileges.

This section describes the authorization methods that Oracle SES supports. The authorization methods prevent search users from accessing documents for which they do not have privileges.

> **See Also:** The Oracle SES admin tutorial at `http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm`

Secure Search Options

Oracle Secure Enterprise Search offers several options for secure search:

- Admin-based Authorization
- Custom Crawler Plug-in
- Query-time Authorization
- Self Service Authorization

Admin-based Authorization

With admin-based authorization, when creating a source, the administrator can specify an authorization policy. This policy governs which users can view each document. Admin-based authorization is based on ACLs. When a source is crawled, each document is stamped with an ACL. When a user enters a search, the result list will only include documents for which the user's credentials match the document's ACL.

> **See Also:** "Authorization and Authentication" on page 4-2 for more information about ACL policies

With the Crawler Plug-in API, the `DocumentAcl` object implements identity-based security. This will use the identity plug-in to validate the users and groups. Only Distinguished Name (DN) and Global User Id (GUID) can be used as the principal of an ACL.

Oracle SES performs ACL duplicate detection. This means that if a crawled document's ACL already exists in the Oracle SES system, then the existing ACL is used to protect the document, instead of creating a new ACL within Oracle SES. This policy reduces storage space and increases performance.

Oracle SES supports only a single LDAP domain. The LDAP users and groups specified in the ACL must belong to the same LDAP domain.

Caution: If ACLs are crawled from sources, then the administrator must ensure that the sources being crawled belong to the same LDAP domain. Otherwise, it is possible that end users are inadvertently granted permission to documents that they should not be able to access.

When secure search is enabled, you could encounter up to a 15 minute delay viewing the private documents. This delay could be due to newly added secure sources or a user/group membership change in the identity management system.

Searches run against a secure-search enabled Oracle SES index are slower than those run against a non-secure-search enabled index. This is because each candidate result could require an ACL evaluation. ACLs are evaluated natively for optimum performance. Nevertheless, the time taken to return results in a secure search varies depending on the number of ACL evaluations that must be made.

Custom Crawler Plug-in

Oracle Secure Enterprise Search provides an API for writing custom crawler plug-ins (or *connectors*) in Java. With this API, you can create a secure crawler plug-in to meet your requirements.

The custom crawler plug-in passes back URLs directly to be indexed. Each URL can be accompanied by an ACL, which restricts the access to that particular document. Alternatively, an ACL can be set in the administration tool for the source.

Authentication credentials can be provided to the plug-in through parameters in the administration tool. The plug-in uses these credentials to access the secure source.

See Also: "Crawler Plug-in API" on page 6-25

Query-time Authorization

Query-time authorization provides another form of filtering. Query-time authorization can be enabled or disabled for Web, file, table, e-mail, mailing list, OracleAS Portal, and user-defined source types from the **Home - Sources - Edit Source** page. It is not available for federated or self-service sources. Query-time authorization can be used with or without ACLs. For example, a source could be stamped with a relatively broad ACL, while query-time authorization could be used to further filter the results.

In query-time authorization, the Oracle SES administrator associates a Java class that is called at run time. The Java class validates each document that is returned in a user query.

Here are the steps involved in query-time authorization:

1. The Oracle SES administrator registers a Java class implementing the `QueryTimeFilter` interface with a source that requires query-time authorization.
2. Oracle SES crawls, collects, and indexes all documents. If ACL stamping has been set up, then it also ACL stamps the documents.
3. At search time, the search result list initially contains all documents accessible under crawl-time ACL policies, unfiltered by query-time user privilege checking.
4. For the top-N results requested by the user, Oracle SES calls the registered Java class, passing in the search request and document information for any documents belonging to the protected source. The Java class returns an integer value for each document indicating if the document should be removed from the result or not.
5. Only items the user is privileged to see are returned to the user in their result list.

Notes for Using Query-time Authorization

- The Browse application is also filtered by the query-time authorization mechanism. The `QueryTimeFilter` class controls which folders are visible to the user, and documents within folders are filtered by the same process as the standard search result list.
- Remember to set the **Hit Count Method** to **Exact count (adjusted for query-time filtering)** on the **Global Settings - Query Configuration** page. If not, then the hit count displayed could be larger than the actual number of documents the user is authorized to view. The page in the administration tool contains other query-time authorization configuration settings you might want to consider.
- If you modify the contents of the jar file containing the `QueryTimeFilter` implementation classes, but do not change the location of the jar file, then you must restart the Oracle SES middle tier using `searchctl restart`. This ensures that the search application picks up your changes and that the Java Virtual Machine does not use a cached version of the class within the old jar file. Restart the middle tier with `searchctl restart`.
- If a `QueryTimeFilter` class is enabled for an OracleAS Portal server, then all of its page group sources are automatically protected by that query-time filter.
- It make take up to five seconds for query-time authorization changes applied in the administration tool to take effect in the Oracle SES search engine. The relevant settings are the following:
  - Enabling a `QueryTimeFilter` class for a source
  - The hit count method
  - The **Query-time Authorization Configuration** settings on the **Global Settings - Query Configuration** page.

    See Also: "Query-time Authorization API" on page 6-31 for more information about implementing a `QueryTimeFilter` Java class

Self Service Authorization

Self service authorization allows end users to enter their credentials needed to access an external content repository. Oracle Secure Enterprise Search crawls and indexes the repository using these credentials to authenticate as the end user. Only the self service user is authorized to see these documents in their search results. Self service authorization works well out of the box, as the crawler appears to be a normally authenticated end user to the content repository.

To set up a self service source, create a template source, defining the target data repository but omitting the credentials needed to crawl. From the search application, an end user can view the **Customize** page and subscribe to a template source by entering their credentials in an input form. A new user-subscribed source is created, along with a copy of the template's schedule. Oracle SES creates an ACL for this user to be applied to the source.

User-subscribed sources are viewable in the **Home - Sources - Manage Template Source** page, and the associated schedules are administered in the **Home - Schedules** page. Any changes applied by the administrator to a template source are dynamically inherited by the associated user-subscribed sources for the next crawl.

The self service option is available for e-mail and Web sources. Self service e-mail sources require the administrator to specify the IMAP server address and the end user to specify the IMAP account user name and password. Self service Web sources are limited to content repositories that use OracleAS Single Sign-On authentication. The administrator specifies the seed URLs, boundary rules, document types, attribute mappings, and crawling parameters, and the end user specifies the single sign-on user name and password.

Crawling of user-subscribed sources is controlled by the administrator. End users will not see any search results for their subscribed source until that source is crawled by the administrator's schedule. Allowing a crawl to automatically launch immediately after an end user subscribes to a source might be useful. However, it makes it possible for users to unintentionally (or intentionally) load the system at inconvenient times.

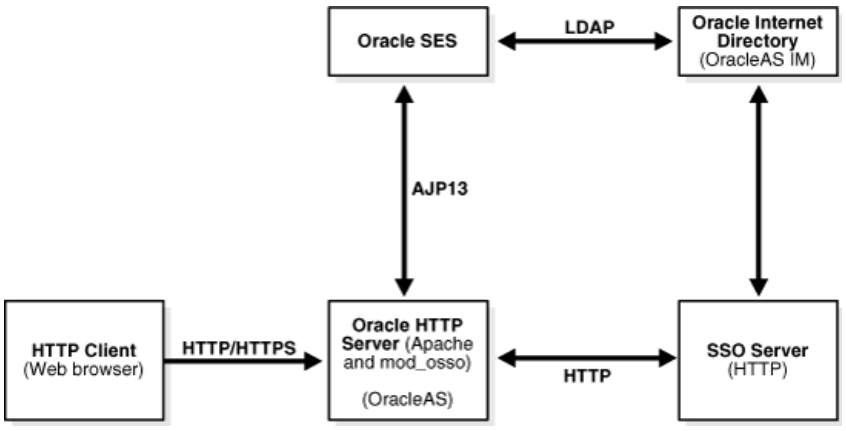
Configuring Oracle Secure Enterprise Search for Oracle Single Sign-On

If you use Oracle Single Sign-On (SSO), then you can configure Oracle SES to use your SSO server for authentication. This section describes the necessary configuration steps.

Note: OracleAS supported versions are 9.0.4 and 10.1.2, with the latest patchsets applied.

`$AS_HOME` refers to the Oracle home directory of the OracleAS middle tier installation.

The following graphic illustrates the configuration:



When using SSO, Oracle SES can be configured to behave in one of the following two ways:

- Mode 2: Allow any user to search public content. After the user logs in, he or she can also search the private `contentIf` submit queries.
- Mode 3: Always require users to log in, even for searching public content.

To SSO-enable Oracle SES, perform the following steps:

1. Front the Oracle SES instance with the Oracle HTTP Server of your OracleAS middle tier. (See "Using mod_oc4j to Front Oracle Secure Enterprise Search with an Oracle HTTP Server" on page 4-12)

   On the OracleAS side, perform the following steps:

2. Configure `mod_osso` to protect the search application with SSO with one of the following steps, depending on the desired secure mode:

   For secure mode 2, add the following lines to `$AS_HOME/Apache/Apache/conf/mod_osso.conf` within the `IfModule` element:

   ```
   <Location /search/query/ssoLogin.jsp>
     require valid-user
     AuthType Basic
   </Location>
   ```

   For secure mode 3, add the following lines to `$AS_HOME/Apache/Apache/conf/mod_osso.conf` within the `IfModule` element:

   ```
   <Location /search/query>
     require valid-user
     AuthType Basic
   </Location>
   ```

3. Restart Oracle HTTP Server. On the OracleAS middle tier host, run the following command:

   ```
   $AS/opmn/bin/opmnctl restartproc process-type=HTTP_Server

   opmnctl: restarting opmn managed processes...
   ```

On the Oracle SES side, perform the following steps:

1. Activate the Oracle Internet Directory identity plug-in on the **Global Settings - Identity Management Setup** page. Log out from the administration tool after activation.
2. Change the secure mode from 1 to 2 or 3. Connect to Oracle SES from the command line using SQL*Plus as the `EQSYS` user, and run one of the following commands. The password for logging into SQL*Plus is the same as the administrator password.

   If only private content in Oracle SES is protected by single-sign on (secure mode 2), then run the following command:

   ```
   SQL> exec eq_adm.set_secure_mode(2);

   PL/SQL procedure successfully completed.

   SQL> exit
   ```

   If both public and private content is protected by single-sign on (secure mode 3), then run the following command:

```
SQL> exec eq_adm.set_secure_mode(3);

PL/SQL procedure successfully completed.

SQL> exit
```

Note: When connecting to the Oracle SES repository using SQL*Plus, make sure that the following environment variables are set correctly:

- `ORACLE_HOME`: The directory where Oracle SES was installed
- `ORACLE_SID`: The search server name specified during installation

Using the password given when installing Oracle SES, run this command:

```
> $ORACLE_HOME/bin/sqlplus eqsys/<password>
```

3. Restart the Oracle SES middle tier using `searchctl restart`.

Using mod_oc4j to Front Oracle Secure Enterprise Search with an Oracle HTTP Server

The Oracle SES middle tier runs in the embedded standalone OC4J. Oracle HTTP Server, on the other hand, contains a module called `mod_oc4j` that allows it to take the role of a frontend HTTP listener to OC4J. HTTP client requests go to the Oracle HTTP Server, which in turn, using `mod_oc4j`, communicates with OC4J through the AJP13 protocol. This makes it possible to front an Oracle SES instance using Oracle HTTP Server.

Note: When using Oracle HTTP Server fronting, Oracle SES allows the Oracle HTTP Server to assert the identity of the current user; therefore, it is of outmost importance to limit this privilege to only trusted Oracle HTTP Server instances. This is done by SSL-protecting the communication between Oracle SES and Oracle HTTP Server.

Special configuration is necessary on both the Oracle SES side and the Oracle HTTP Server side.

On the Oracle SES side, do the following:

1. Edit the `$ORACLE_HOME/oc4j/j2ee/OC4J_SEARCH/config/http-web-site.xml` file. In the element `web-site`, change the attribute `protocol` from `"http"` to `"ajp13"`:

   ```
   <web-site ... protocol="ajp13" ... >
   ```

2. Enable SSL. (See "SSL and HTTPS Support in Oracle Secure Enterprise Search" on page 4-13)
3. Restart the Oracle SES middle tier using `searchctl restart`.

Next, on the Oracle HTTP Server's middle tier, perform the following steps:

1. Configure Oracle HTTP Server to forward requests to the Oracle SES middle tier. Edit the `$AS_HOME/Apache/Apache/conf/mod_oc4j.conf` file. Within the `IfModule` element, add the following line:

   ```
   Oc4jMount  /search/*   ajp13://<sesHost>:<sesPort>
   ```

where <sesHost> and <sesPort> are the host name and middle tier port number of the Oracle SES instance

2. Enable SSL. (See "Enabling SSL in Oracle HTTP Server's mod_oc4j Module" on page 4-17)
3. Restart Oracle HTTP Server. On the OracleAS middle tier host, run the following command:

```
$AS_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

At this point, to access the Oracle SES middle tier you need to go through the Oracle HTTP Server. In other words, for the Oracle SES URLs you now have to use the host and port of the Oracle HTTP Server. The original URLs are no longer accessible.

SSL and HTTPS Support in Oracle Secure Enterprise Search

Oracle SES can crawl HTTPS-based URLs, and the Oracle SES middle tier can be configured to support HTTPS-based access. HTTPS is nothing more than HTTP running over a secure socket layer (SSL).

Understanding SSL

SSL is an encryption protocol for securely transmitting private content on the internet. With SSL, two parties can establish a secure data channel. SSL uses a cryptographic system that uses two keys to encrypt data: a public key and a private key. Data encrypted with the public key can only be decrypted using the private key, and vice versa.

In SSL terms, the party that initiates the communication is considered the client. During the SSL handshake, authentication between the two parties occurs. The authentication can be one sided (server authentication only) or two sided (server and client authentication).

Server authentication is more common. It happens every time a Web browser accesses a URL that starts with HTTPS. Thanks to server authentication, the client can be certain of the server's identity and can trust that it is safe to submit to the server secure data, such as login username and password.

The following list defines some common terms related to SSL:

- **Keystore**: A repository that includes the following:
  - Certificates identifying trusted entities. When a keystore only contains certificates of trusted entities it can be called a *truststore*
  - Private-key and the matching certificate. This certificate is sent as a response to SSL authentication challenges.
- **Certificate**: A digital identification of an entity that contains the following:
  - SSL public key of the server
  - Information about the server
  - Expiration date
  - Digital signature by the issuer of the certificate used to verify the authenticity of the certificate
- **Certificate authority (CA)**: A well known and trusted entity (for example, VeriSign or Thawte). CAs are usually the issuers of other certificates

- **Root certificate**: A self-signed certificate where the issuer is the same entity as what the certificate represents. CA certificates are generally root certificates.
- **Certificate chain**: This chain is comprised of the certificate, its issuer, the issuer of the issuer, and so on, all the way to the root certificate. If one certificate in the chain is trusted (that is, it is in the keystore), then the rest of the certificate can be verified for authenticity. This makes it possible for a keystore to contain only a few well-known and trusted root certificates from which most other certificates originate.

Every SSL connection starts with the SSL handshake. There is quite a bit involved in the SSL handshake. This section describes the basic steps involved in it:

1. The client contacts the server to establish a SSL connection.
2. The server looks in its keystore for its own SSL certificate and sends it back to the client.
3. The client checks its keystore to see if it trusts the server or any of the entities in the server's certificate chain. If not, then the handshake is aborted. Otherwise, the client positively identifies the server and deems it trusted. The expiration date of the certificate is also checked, and the name on the certificate is matched against the domain name of the server.
4. If the server is configured to require client authentication, then the server will ask the client to identify itself, so the mirror image of steps 2 and 3 will take place.
5. Session keys are generated. From now on, session keys are used for encrypting exchanged data.

SSL in Oracle Secure Enterprise Search

For SSL support, Oracle SES uses JSSE, a highly customizable SSL package included in Sun Microsystem's J2SE.

Oracle SES uses SSL for many operations, in some acting as the SSL client, and others acting as the SSL server.

Examples when Oracle SES acts as the SSL client:

- The crawler accesses a data repository that uses SSL (for example, HTTPS Web sites)
- The form registration wizard in the administration tool accesses HTTPS URLs
- Oracle SES federates queries to other SSL-enabled search services (for example, an SSL-enabled Oracle SES instance)

An example of when Oracle SES acts as the SSL server:

- The Oracle SES middle tier, configured to use SSL, responds to HTTPS or AJP13 requests.

Managing the Keystore

Out of the box, Oracle SES uses the default keystore in J2SE: `$ORACLE_HOME/jdk/jre/lib/security/cacerts`. The keystore's password is `changeit`. This keystore is populated with the root certificates representing the well known certificate authorities. (Most SSL-enabled Web sites use certificates that originate or chain from these main root certificates.)

See Also:

http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html

Depending on requirements, the keystore might still need maintenance. For example:

- If one of the main root certificates expires, then it will need to be replaced by a new issue.
- If Oracle SES needs to trust another SSL-enabled peer whose certificate does not originate from one of the root certificates, then the peer's certificate, or one from its chain, needs to be added to the keystore.
- To enable SSL in the Oracle SES middle tier, Oracle SES must act as an SSL server, and that calls for the keystore to contain a private key and the corresponding certificate with the public key. (The same holds true for the SSL client role where the server requires client side SSL authentication.)

Maintenance of the keystore can be done using Sun's keytool program, which ships with J2SE. (You can find this tool under `$ORACLE_HOME/jdk/bin`). Third-party keytool GUI wrapper programs are available.

See Also:

- "Understanding SSL" on page 4-13
- http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html for detailed instructions on how to add, remove or update certificates, generate keys, and create new keystores with keytool

Configuring Oracle Secure Enterprise Search to Require SSL

Clients (Web browsers, Web service clients, and so on) interact with Oracle SES directly using HTTP. If Oracle SES is fronted by Oracle HTTP Server, as it is needed for SSO support, then HTTP clients interact with Oracle HTTP Server, and Oracle HTTP Server forwards the requests to Oracle SES using AJP13.

Note: When Oracle SES is configured to use the AJP13 protocol (that is, when Oracle SES is fronted by an Oracle HTTP Server), it is strongly recommended that Oracle SES be configured to require SSL with client-side authentication for communication with the Oracle HTTP Server. Furthermore, a keystore other than the default one should be used. While the default keystore contains the trusted certificates of all the major Certificate Authorities, the keystore used for the AJP13 SSL channel must contain ONLY Oracle SES's own certificate and the trusted certificate of the fronting Oracle HTTP Server.

The communication channel between the client and Oracle SES is (by default) not SSL-enabled and not encrypted. To protect this channel with SSL, follow these steps:

1. Shut down the middle tier with `$ORACLE_HOME/bin/searchctl stop`.
2. Change to directory `$ORACLE_HOME/oc4j/j2ee/OC4J_SEARCH/config`.
3. Edit the `http-web-site.xml` file.

   To the `<web-site>` element, add the attribute `secure="true"`. For example:

```
<web-site ... protocol="http" secure="true"... >
   ...
</web-site>
```

To the web-site element, add the <ssl-config> subelement and its keystore and keystore-password attributes, which specify the directory path and password for the keystore. For example:

```
<web-site ... secure="true" ... >
   ...
   <ssl-config keystore="$ORACLE_HOME/jdk/jre/lib/security/cacerts"
               keystore-password="changeit"
               needs-client-auth="false" />
</web-site>
```

To the <web-app> elements, add the attribute shared="true". For example:

```
<web-app application="search_query" . . . shared="true" />
```

If the protocol attribute is set to AJP13 (that is, if Oracle SES is fronted with Oracle HTTP Server), then use SSL to control which Oracle HTTP Servers are allowed to front Oracle SES. To do this, configure Oracle SES to require client-side SSL authentication, and make sure that the keystore specified in the <ssl-config> element only contains the SSL certificate of the fronting Oracle HTTP Server.

For example:

a. In the <ssl-config> element added earlier, set the attribute keystore="./cacerts" and set needs-client-auth="true".

b. From the administrator of the fronting Oracle HTTP Server, get its SSL certificate and import it into the keystore specified in the <ssl-config> element. For example:

```
$ORACLE_HOME/jdk/bin/keytool -import -keystore ./cacerts -trustcacerts
-alias myOHS -file <path to the file containing the Oracle HTTP Server's
certificate (for example, "/temp/ohs.cer")>
```

If the keystore specified using the -keystore argument does not already exist, then a new empty keystore will be created. You will be asked for the keystore password. The default keystore password is changeit. You will be asked for confirmation to import the certificate into your specified keystore.

Note: If Oracle SES is fronted with Oracle HTTP Server, and Oracle SES is configured to require SSL for its communication with Oracle HTTP Server, then Oracle HTTP Server's mod_oc4j module also needs to be configured for SSL. For more information, see "Enabling SSL in Oracle HTTP Server's mod_oc4j Module" on page 4-17 or see the OracleAS documentation.

---

4. Using keytool, add a key/certificate pair to the keystore specified in the <ssl_config> element.

- The name on the certificate should be the host on which Oracle SES is running.
- The key algorithm should be "RSA" (that is, use the keytool option "-keyalg RSA")

- If the certificate is not issued or signed by a well-known CA, then the certificate, or one in its chain, must be added to the keystore of every client that will communicate with the Oracle SES instance.

Suggestion: Backup the keystore before modifying it.

For example:

```
$ORACLE_HOME/jdk/bin/keytool -genkey -keyalg RSA -alias oses
-keystore <path to the keystore as specified in the keystore attribute
of the <ssl_config> element>
```

You will be asked a series of questions. When asked, `"What is your first and last name?"`, specify the host name of the Oracle SES machine. For example, `myoses.us.oracle.com`.

5. If you are using a certificate that is not signed by a well-known CA (the earlier example creates a self-signed certificate), then export the Oracle SES certificate so that it can be imported as a trusted certificate for clients:

```
$ORACLE_HOME/jdk/bin/keytool -export -alias oses
-keystore <path to keystore>
-file <path to file for exported certificate, for example /temp/oses.cer>
```

6. Start the Oracle SES middle tier with `$ORACLE_HOME/bin/searchctl start`.

Enabling SSL in Oracle HTTP Server's mod_oc4j Module

Previous sections described the configuration steps on the Oracle SES side of the communications channel. This section describes the configuration steps for the Oracle HTTP Server.

Configuring the Oracle HTTP Server to require SSL for its AJP13 communication channel with Oracle SES does not change the manner in which Web browsers or other HTTP clients communicate with the Oracle HTTP Server.

The following steps SSL-enable `mod_oc4j`:

1. Set up an Oracle Wallet to be used as an SSL keystore by the `mod_oc4j` module. The Oracle Wallet must contain a valid key-cert pair. If such a wallet exists, then skip to step 2.

   a. Create a new wallet using Oracle Wallet Manager (`$OH/bin/owm`). You will be asked to specify the directory in which to hold the wallet and the password for the wallet. Under the **Wallet** menu, turn on the **Auto Login** option.

   b. Create a key-cert pair (that is, a user certificate). Note that the CN part of the DN of the subject of the user certificate needs to set to the machine host name. Also, note that the DN is case sensitive, so make sure to use the same case consistently.

      If the Oracle HTTP Server version is 10.1.2 or later, then you can do this using the `orapki` utility:

```
$AS_HOME/bin/orapki wallet add
-wallet <path to directory containing the wallet>
-dn <DN of the subject
(for example, CN=myhost.oracle.com,OU=oses,O=oracle,ST=ca,C=US)>
-keysize 1024 -self_signed -validity 720
```

      If the Oracle HTTP Server version is earlier than 10.1.2, then you have to create a certificate request using the Oracle Wallet Manager, have the certificate

request signed by a CA, and then use Oracle Wallet Manager to import the CA signed certificate back into the Oracle Wallet.

The **Operations** menu lists the options to create a certificate request and then export that request. Export the request to a file (for example, `clientapp.crs`).

To get the certificate signed you have three options:

- Send the certificate request to a well known CA, such as VeriSign, to have it signed. A fee is charged for this. If you plan to use the same Oracle Wallet and certificate for HTTPS enabling your production Oracle HTTP Server, then this method is recommended.

- If you are using OracleAS Certificate Authority, then you can use it to sign the certificate request.

- You can use OpenSSL to create a CA and use it to have your certificate request signed. For instructions on how to do this, see "OpenSSL as a Certificate Authority" on page 4-18.

After you get your certificate request signed, import the response into the Oracle Wallet.

See Also: "Managing Wallets and Certificates" in the *Oracle Application Server Administrator's Guide* for more information on Oracle Wallets and the `orapki` utility

2. Exchange trusted certificates with the Oracle SES Server which is to be fronted by this Oracle HTTP Server. Use the Oracle Wallet Manager to import/export certificates to and from the Oracle Wallet and use the Java keytool for the Oracle SES keystore.

   When importing a certificate, if the certificate is not self-signed, then before importing it you must import the certificates in its chain.

3. Enable SSL in the `mod_oc4j` module (if not already enabled).

   Navigate to the `$AS_HOME/Apache/Apache/conf` directory and edit the `mod_oc4j.conf` file by adding the following directives within the `IfModule` element:

   ```
   Oc4jEnableSSL On
   Oc4jSSLWalletFile <path to the DIRECTORY containing the oracle wallet>
   ```

   After `mod_oc4j` has been configured to use SSL, it will only be able to front AJP13 servers that also have been SSL-enabled.

4. Restart Oracle HTTP Server. On the OracleAS middle tier host, run the following command:

   ```
   $AS_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
   ```

OpenSSL as a Certificate Authority

OpenSSL is an open source SSL toolkit that can be used to create a CA and use the CA to sign other certificate requests.

1. Install OpenSSL
2. Setup the OpenSSL directory structure:

```
mkdir makecert
cd makecert
mkdir demoCA
cd demoCA
mkdir certs crl newcerts private
touch index.txt
echo "01" > serial
cd ..
```

3. Create the CA (self signed key-cert pair):

```
openssl genrsa -out ca.key 1024
openssl req -new -x509 -key ca.key -out demoCA/cacert.pem
```

At this point, you are ready to sign SSL certificate signing requests generated by tools like keytool or Oracle Wallet Manager. Assuming that the certificate signing request is `clientapp.crs`, run the following commands:

```
openssl ca -keyfile ca.key -in clientapp.crs -out clientapp.pem
openssl x509  -outform DER -in clientapp.pem -out clientapp.der
```

The first command signs the certificate, and the second command transforms the signed certificate into the DER format.

The signed certificate `clientapp.der` is ready to be imported in the keystore from which the certificate signing request was generated.

Note: Before importing `clientapp.der`, you must first import the certificate of its signer: `demoCA/cacert.pem`.

Security in a Federated Search Environment

To perform secure search in a federated search environment, various aspects of security must be considered. See "Setting Up Secure Federated Search" on page 5-4.

5

Oracle Secure Enterprise Search Advanced Information

This chapter contains the following topics:

Tips for Using Table Sources

Oracle Secure Enterprise Search can crawl table sources in an Oracle database. To crawl non-Oracle databases, you must create a view in an Oracle database on the non-Oracle table. Then create the table source on the Oracle view. Oracle SES accesses databases using database links.

Limitations with Table Sources

- Oracle SES cannot crawl tables inside the Oracle SES database.
- Only one table or view can be specified for each table source. If data from more than one table or view is required, then first create a single view that encompasses all required data.
- Table column mappings cannot be applied to LOB columns.
- The following data types are supported for table sources: `BLOB`, `CLOB`, `CHAR`, `VARCHAR`, `VARCHAR2`.

Limitations with Database Links

- If the text column of the base table or view is of type `BLOB` or `CLOB`, then the table must have a `ROWID` column. A table or view might not have a `ROWID` column for various reasons, including the following:
  - A view is comprised of a join of one or more tables.
  - A view is based on a single table using a `GROUP BY` clause.

  The best way to know if a table or view can be safely crawled by Oracle SES is to check for the existence of the `ROWID` column. To do so, run the following SQL statement against that table or view using SQL*Plus: `SELECT MIN(ROWID) FROM <table or view name>;`
- The base table or view cannot have text columns of type `BFILE` or `RAW`.

Tips for Using File Sources

This section contains the following:

- Crawling File Sources with Non-ASCII
- Crawling File Sources with Symbolic Links
- Crawling File URLs

Crawling File Sources with Non-ASCII

For file sources to successfully crawl and display multibyte environments, the locale of the machine that starts the Oracle SES server must be the same as the target file system. This way, the Oracle SES crawler can "see" the multibyte files and paths.

If the locale is different in the installation environment, then Oracle SES should be restarted from the environment with the correct locale. For example, for a Korean environment, either set `LC_ALL` to `ko_KR` **or** set both `LC_LANG` **and** `LANG` to `ko_KR.KSC5601`. Then run `searchctl restartall` from either a command prompt on Windows or an xterm on UNIX.

Crawling File Sources with Symbolic Links

When crawling file sources on UNIX, the crawler will resolve any symbolic link to its true directory path and enforce the boundary rule on it. For example, suppose directory `/tmp/A` has two children, `B` and `C`, where `C` is a link to `/tmp2/beta`. The crawl will have the following URLs:

- /tmp/A
- /tmp/A/B
- /tmp2/beta
- /tmp/A/C

If the boundary rule is /tmp/A, then /tmp2/beta will be excluded. The seed URL is treated as is.

Crawling File URLs

If a file URL is to be used "as is", without going through Oracle SES for retrieving the file, then "file" in the URL should be upper case "FILE". For example, FILE://localhost/... "As is" means that when a user clicks on the search link of the document, the browser will try to use the specified file URL on the client machine to retrieve the file. Without that, Oracle SES uses this file URL on the server machine and sends the document through HTTP to the client machine.

Tips for Using Mailing List Sources

The Oracle SES crawler is IMAP4 compliant. To crawl mailing list sources, you need an IMAP e-mail account. It is recommended to create an e-mail account that is used solely for Oracle SES to crawl mailing list messages. The crawler is configured to crawl one IMAP account for all mailing list sources. Therefore, all mailing list messages to be crawled must be found in the Inbox of the e-mail account specified on this page. This e-mail account should be subscribed to all the mailing lists. New postings for all the mailing lists will be sent to this single account and subsequently crawled.

Messages deleted from the global mailing list e-mail account are not removed from the Oracle SES index. In fact, the mailing list crawler itself will delete messages from the IMAP e-mail account as it crawls. The next time the IMAP account for mailing lists is crawled, the previous messages will no longer be there. Any new messages in the account will be added to the index (and also consequently deleted from the account). This keeps the global mailing list IMAP account clean. The Oracle SES index serves as a complete archive of all the mailing list messages.

Tips for Using OracleAS Portal Sources

- An OracleAS Portal source name cannot exceed 35 characters.
- URL boundary rules are not enforced for URL items. A URL item is the metadata that resides on the OracleAS Portal server. Oracle SES does not touch the display URL or the boundary rules for URL items.

Tips for Using User-Defined Sources

If a plug-in is to return file URLs to the crawler, then the file URLs must be fully qualified. For example, file://localhost/.

Also, if a file URL is to be used "as is" without going through Oracle SES for retrieving the file, then "file" in the URL should be upper case "FILE". For example, FILE://localhost/...

See Also: "Crawling File URLs" on page 5-3

Tips for Using Federated Sources

Oracle SES provides the capability of searching multiple Oracle SES instances with their own document repositories and indexes. It provides a unified framework to search the different document repositories that are crawled, indexed, and maintained separately. Federated search allows a single query to be run across all Oracle SES instances. It aggregates the search results to show one unified result list to the user. User credentials are passed along with the query so that each remote (that is, slave) Oracle SES application can authenticate the user against its own document repository.

Create a federated source on the **Home - Sources** page of the Oracle SES administration tool.

Notes:

- The Oracle SES federator caches the federator configuration (that is, all federation-related parameters including federated sources). As a result, any change in the configuration will take effect within 0 to 5 minutes.
- Oracle SES supports 2-tier federated search. Federation of 3-tier or more is not currently supported.

See Also: "Setting Up Secure Federated Search" on page 5-4 if the federated source will be searching private content

Federated Search Characteristics

- Federated search can improve performance by distributing query processing on multiple machines. It can be an efficient way to scale up search service by adding a cluster of Oracle SES instances.
- The federated search performance depends on the network topology and throughput of the entire federated Oracle SES environment.

Federated Search Limitations

- There is a size limit of 200KB for the cached documents existing on the remote Oracle SES instance to be displayed on the master node.
- For infosource browse, if the source hierarchies for both local and federated sources under one source group start with the same top level folder, then only one of the hierarchies is available for browse.
- On the master federated Oracle SES instance, there is no direct access to documents on the remote Oracle SES instance through the display URL in the search result list. Only the cached version of documents is accessible. **Exception**: There *is* direct access for Web source and OracleAS Portal source documents.

Setting Up Secure Federated Search

Secure federated search enables searching secure content across distributed Oracle SES instances. An end user is authenticated to the Oracle SES master instance. Along with querying the secure content in its own index, the master instance federates the query to each of the remote (that is, slave) Oracle SES instances on behalf of the authenticated end user. This mechanism necessitates propagation of user identity between the Oracle SES instances. In building a secure federated search environment, an important

consideration is the secure propagation of user identities between the SES instances. This section explains how Oracle SES performs secure federation.

Federation Trusted Entities

When performing a secure search on a remote Oracle SES instance, the master instance must pass the identity of the logged in user to the remote instance. If the remote instance trusts the master instance, then the master instance can proxy as the end user. To establish this trust relationship, Oracle SES instances should exchange some secret. This secret is exchanged in the form of a *trusted entity*. A trusted entity consists of two values: entity name and entity password. Each Oracle SES instance can have one or more trusted entities that it can use to participate in secure federated search. (A trusted entity is also referred to as a proxy user.)

Create trusted entities on the **Global Settings - Federation Trusted Entities** page of Oracle SES administration tool.

An Oracle SES instance can connect to an identity management (IDM) system for managing users and groups. An IDM system can be an LDAP compliant directory, such as Oracle Internet Directory or Active Directory.

Each trusted entity can be authenticated by either an IDM system or by the Oracle SES instance directly, independent of an IDM system. For authentication by an IDM system, check the box **Use Identity Plug-in for authentication** when creating a trusted entity. In this case, the entity password is not required. This is useful when there is a user configured in the IDM system that can be used for proxy authentication. Make sure that the entity name is the name of the user that exists in the IDM system and is going to be used as the proxy user.

For authentication of the proxy user by Oracle SES, clear (uncheck) the box **Use Identity Plug-in for authentication** when creating a trusted entity. Then use any name and password pair to create a trusted entity.

To perform secure federated search, both Oracle SES instances involved in the federation must have identity plug-ins registered. The identity plug-ins may or may not talk to the same IDM system. Carefully specify the following parameters under the section **Secure Federated Search** when creating a federated source on the master Oracle SES instance:

- **Remote Entity Name**: This is the name of the federation trusted entity on the remote Oracle SES instance provided by the administrator of the remote Oracle SES instance.
- **Remote Entity Password**: This is the password of the federation trusted entity on the remote Oracle SES instance provided by the administrator of the remote Oracle SES instance.
- **Search User Attribute**: This attribute identifies, and is used to authenticate, a user on the remote Oracle SES instance. This parameter is optional, except in the case where the master and remote SES instances use different authentication attributes to identify end users. The master and remote Oracle SES instances can use different authentication attributes to identify or authenticate end users. (For example, on the master instance, an end user can be identified by user name; on the remote instance, the end user can be identified by e-mail address.)

  The identity plug-in registered on the master instance should be able to map the user identity to this attribute based on the authentication attribute used during the registration of the identity plug-in. If this attribute is not specified during creation of the federation source, then the user identity on the master instance is used to search on the remote Oracle SES instance.

Note: If these parameters are not specified during the creation of the federated source, then the federated source is treated as a public source (that is, only public content is available to the search users).

- **Secure Oracle HTTP Server-Oracle SES channel**: Because any Oracle HTTP Server can potentially connect to the AJP13 port on the Oracle SES instances and masquerade as a specific person, either the channel between the Oracle HTTP Server and the Oracle SES instance must be SSL-enabled or the entire Oracle HTTP Server and Oracle SES instance machines must be protected by a firewall.

Notes:

- In a secure federated search environment, the master instance might or might not be using single sign-on (SSO). However, the Web service should not be behind SSO. The remote Oracle SES instance cannot use SSO to protect all content, but it can use SSO to protect *private* content.
- Oracle strongly recommends that you SSL-protect the channel between Oracle HTTP Server and Oracle SES for secure content. The remote instance should be SSL-enabled, or you should be able to access the Web service using HTTPS.

See Also: "Configuring Oracle Secure Enterprise Search for Oracle Single Sign-On" on page 4-10

Tips for Using Oracle Calendar Sources

Oracle Calendar sources are certified with Oracle Calendar release 10.1.2.

Oracle recommends creating one source group for *archived* calendar data and another source group for *active* calendar data. One Calendar connector instance for the archived source can run less frequently, such as every week or month. This source should cover all history. A separate connector instance for the active source can run daily for only the most recent period.

Setting Up Secure Oracle Calendar Sources

The Oracle SES instance and the Oracle Calendar instance must be connected to the same Oracle Internet Directory system. Follow these steps to set up a secure Oracle Calendar source:

1. Activate the Oracle Internet Directory identity plug-in for the Oracle Calendar instance. This is done on the **Global Settings - Identity Management Setup** page in the Oracle SES administration tool.
2. Use the following LDIF file to create an *application entity* for the plug-in. (An application entity is a data structure within LDAP used to represent and keep track of software applications accessing the directory via an LDAP client.)

```
$ORACLE_HOME/bin/ldapmodify -h oidHost -p OIDPortNumber -D OIDadmin -w password
-f  calPlugin.ldif
```

Where `$ORACLE_HOME` is the Oracle Calendar infrastructure installation and `calPlugin.ldif` is the current directory.

This defines the entity that will be used for the plug-in: `orclapplicationcommonname=ocscalplugin,cn=oses,cn=product, cn=oraclecontext`. The entity will have the password `welcome1`.

> **See Also:** Appendix D, "LDIF Files" to view the `calPlugin.ldif` file

3. Create a Calendar source on the **Home - Sources** page.

Table 5–1 Calendar Source Parameters

Parameter	Value
Calendar server	http://*host name*:*port*
Application entity name	*name*
Application entity password	welcome1
OID server hostname	*host name*
OID server port	389
OID server SSL port	636
OID server ldapbase	dc=us,dc=oracle,dc=com
OID login attribute	uid
User query	(objectclass=ctCalUser)
Past days	30
Future days	60
Rollover	true

Tips for Using Oracle Content Database Sources

Oracle Content Database and Oracle Content Services are the same product. This section uses the product name Oracle Content Database to mean Oracle Content Database *and* Oracle Content Services. Oracle Content Database sources are certified with Oracle Content Database release 10.2 and Oracle Content Services release 10.1.2.3.

Limitations with Oracle Content Database Sources

- Oracle SES currently does not index Oracle Content Database Categories; that is, custom metadata such as ProjectNumber, Client, or Project Manager.
- The administrator account used by the Oracle Content Database source must have the `ContentAdministrator` role on the site that is being crawled and indexed. Also, end-users searching documents in Oracle Content Database must have the `GetContent` and `GetMetadata` permissions.
- By default, Oracle Content Database has a limit of three concurrent requests (simultaneous operations) for each user. However, Oracle SES has a default of five concurrent crawler threads. When crawling Oracle Content Database, only three of the five threads can successfully crawl, which causes the crawl to fail.

  Workaround: For an Oracle Content Database source, change the **Number of Crawler Threads** on the **Home - Sources - Crawling Parameters** page to a value less than or equal to three.

Or, modify the Oracle Collaboration Suite configuration in Oracle Enterprise Manager to allow more than three concurrent requests. For example:

1. Access the Enterprise Manager page for the Collaboration Suite Midtier. For example: `http://machine.domain:1156/`.
2. Click the Oracle Collaboration Suite midtier standalone instance name. For example: `ocsapps.machine.domain`.
3. In the **System Components** table, click **Content**.
4. From **Administration**, click **Node Configurations**.
5. In the **Node Configurations** table, click **HTTP_Node**. For example: `ocsapps.machine.domain_HTTP_Node`.
6. On **Properties**, change the value for **Maximum Concurrent Requests Per User**. Enter a value larger than or equal to the number of crawling threads used by Oracle SES. This value is listed on the **Global Settings - Crawler Configuration** page.

Setting Up Secure Oracle Content Database Sources

The Oracle SES instance and the Oracle Content Database instance must be connected to the same Oracle Internet Directory system. The groups in Oracle Content Database must also be synchronized with Oracle Internet Directory. Follow these steps to set up a secure Oracle Content Database source:

1. Read "Limitations with Oracle Content Database Sources" on page 5-7 and confirm that the number of crawler threads does not exceed the available per-user concurrent connection settings in Oracle Content Database
2. Activate the **Secure Enterprise Search Group Agent**. Oracle Content Database uses this agent to synchronize groups into Oracle Internet Directory, so that they can be used by Oracle SES.

   This agent is deactivated by default. Activate it by modifying the node configuration that corresponds to the node where you want to run the agent. To do this, follow these steps:

   a. Connect to the Oracle Collaboration Suite Control and go to the **Content Database Home** page. From a Web browser, connect to the Enterprise Manager port, which is typically http://servername:1156. Log on as `ias_admin` with the password provided during the Oracle Collaboration Suite or Oracle Application Server installation. Choose the correct cluster name (which will be for APPS, not INFRASTRUCTURE). The names vary depending the installation. Under **System Components**, choose **Content**.
   b. In the Administration section, click **Node Configurations**.
   c. On the **Node Configurations** page, click the name of the node configuration you want to change.
   d. In the **Servers** section, click **Activate/Deactivate**.
   e. Move the **Secure Enterprise Search Group Agent** from the **Inactive Servers** list to the **Active Servers** list.
   f. Click **OK** on the **Activate/Deactivate Servers** page.
   g. Click **OK** on the **Edit Node** page.
   h. Return to the **Content Database Home** page and restart the node.

The crawler authenticates as an administrator user who has privilege to read all contents of all folders in the Oracle Content Database repository. This uses the service-to-service mechanism of passing in a trusted application entity name and password along with the admin user name.

3. Activate the Oracle Internet Directory identity plug-in for the Oracle Content Database instance. This is done on the **Global Settings - Identity Management Setup** page in the Oracle SES administration tool.
4. Use the following LDIF file to create an *application entity* for the plug-in. (An application entity is a data structure within LDAP used to represent and keep track of software applications accessing the directory via an LDAP client.)

```
$ORACLE_HOME/bin/ldapmodify -h oidHost -p OIDPortNumber -D OIDadmin -w password
-f  csPlugin.ldif
```

Where `$ORACLE_HOME` is the Oracle Content Database infrastructure installation and `csPlugin.ldif` is the current directory.

This defines the entity that will be used for the plug-in: `orclapplicationcommonname=ocscsplugin, cn=ifs,cn=products,cn=oraclecontext`. The entity will have the password `welcome1`.

> **See Also:** Appendix D, "LDIF Files" to view the `csPlugin.ldif` file

5. Create an Oracle Content Database source on the **Home - Sources** page.

Table 5–2 Content Database Source Parameters

Parameter	Value
Oracle Content Database URL	http://*host name*:*port*
Starting paths	/
Depth	-1
Oracle Content Database admin user	orcladmin
Entity name	`orclapplicationcommonname=ocscsplugin, cn=ifs,cn=products,cn=oraclecontext`
Entity password	welcome1
LDAP group base	`cn=CSGroups,cn=groups,dc=us,dc=oracle,dc=com`
Crawl only	false
Use e-mail for authorization	false

Tuning Crawl Performance

Your Web crawling strategy can be as simple as identifying a few well-known sites that are likely to contain links to most of the other intranet sites in your organization. You could test this by crawling these sites without indexing them. After the initial crawl, you have a good idea of the hosts that exist in your intranet. You could then define separate Web sources to facilitate crawling and indexing on individual sites.

However, the process of discovering and crawling your organization's intranet, or the Internet, is generally an interactive one characterized by periodic analysis of crawling results and modification to crawling parameters. For example, if you observe that the

crawler is spending days crawling one Web host, then you might want to exclude crawling at that host or limit the crawling depth.

This section contains the most common things to consider to improve crawl performance:

- Register a Proxy
- Check Boundary Rules
- Check Dynamic Pages
- Check Crawler Depth
- Check Robots.txt Rule
- Check Duplicate Pages
- Check Redirected Pages
- Check URL Looping
- What to do Next

> **See Also:** "Monitoring the Crawling Process" on page 3-8 for more information on crawling parameters

Register a Proxy

By default, Oracle SES is configured to crawl Web sites in the intranet. In other words, crawling internal Web sites requires no additional configuration. However, to crawl Web sites on the Internet (also referred to as external Web sites), Oracle SES needs the HTTP proxy server information. See the **Global Settings - Proxy Settings** page.

If the proxy requires authentication, then enter the proxy authentication information on the **Global Settings - Authentication** page.

Check Boundary Rules

The seed URL you enter when you create a source is turned into an inclusion rule. For example, if www.example.com is the seed URL, then Oracle SES creates an inclusion rule that only URLs containing the string www.example.com will be crawled.

However, suppose that the example Web site includes URLs starting with www.exa-mple.com or ones that start with example.com (without the www). Many pages have a prefix on the site name. For example, the investor section of the site has URLs that start with investor.example.com.

Always check the inclusion rules before crawling, then check the log after crawling to see what patterns have been excluded.

In this case, you might add www.example.com, www.exa-mple.com, and investor.example.com to the inclusion rules. Or you might just add example.

To crawl outside the seed site (for example, if you are crawling text.us.oracle.com, but you want to follow links outside of text.us.oracle.com to oracle.com), consider removing the inclusion rules altogether. Do so carefully. This could lead the crawler into many, many sites.

Notes for File Sources

1. For file sources, if no boundary rule is specified, then crawling is limited to the underlying file system access privileges. Files accessible from the specified seed file URL will be crawled, subject to the default crawling depth. The depth, which

is 2 by default, is set on the **Global Settings - Crawler Configuration** page. For example, if the seed is `file://localhost/home/user_a/`, then the crawl will pick up all files and directories under `user_a` with access privileges. It will crawl any documents in the directory `/home/user_a/level1` due to the depth limit. The documents in the `/home/user_a/level1/level2` directory are at level 3.

2. The file URL can be of UNC (universal naming convention) format. The UNC file URL has the following format:
`file://localhost///<LocalMachineName>/<SharedFolderName>`.

   For example, `\\stcisfcr\docs\spec.htm` should be specified as `file://localhost///stcisfcr/docs/spec.htm`.

3. On some machines, the path or file name could contain non-ASCII and multibyte characters. URLs are always represented using the ASCII character set. Non-ASCII characters are represented using the hex representation of their UTF-8 encoding. For example, a space is encoded as %20, and a multibyte character can be encoded as %E3%81%82.

   For file sources, spaces can be entered in simple (not regular expression) boundary rules. Oracle SES automatically encodes these URL boundary rules. If (Home Alone) is specified, then internally it is stored as (Home%20Alone). Oracle SES does this encoding for the following:

   - File source simple boundary rules
   - Test URL strings
   - File source seed URLs

   Note: Oracle SES does not alter the rule if it is a regular expression rule. It is the administrator's responsibility to make sure that the regular expression rule specified is against the encoded file URL. Spaces are not allowed in regular expression rules.

Check Dynamic Pages

Indexing dynamic pages can generate an excessive number of URLs. From the target Web site, manually navigate through a few pages to understand what boundary rules should be set to avoid crawling identical pages.

Check Crawler Depth

Setting the crawler depth very high (or unlimited) could lead the crawler into many sites. Without boundary rules, 20 will probably crawl the whole WWW from most locations.

Check Robots.txt Rule

You can control which parts of your sites can be visited by robots. If robots exclusion is enabled (default), then the Web crawler traverses the pages based on the access policy specified in the Web server `robots.txt` file.

The following sample `/robots.txt` file specifies that no robots should visit any URL starting with `/cyberworld/map/` or `/tmp/` or `/foo.html`:

```
# robots.txt for http://www.example.com/

User-agent: *
Disallow: /cyberworld/map/
Disallow: /tmp/
Disallow: /foo.html
```

If the Web site is under the user's control, then a specific robots rule can be tailored for the crawler by specifying the Oracle SES crawler plug-in name "User-agent: Oracle Secure Enterprise Search." For example:

```
User-agent: Oracle Secure Enterprise Search

Disallow: /tmp/
```

The robots `meta` tag can instruct the crawler to either index a Web page or follow the links within it. For example:

```
<meta name="robots" content="noindex,nofollow">
```

Check Duplicate Pages

If Oracle SES thinks a page is identical to one it has seen before, then it will not index it. If the page is reached through a URL that Oracle SES has already processed, then it will not index that either.

Check Redirected Pages

The crawler crawls only redirected pages. For example, a Web site might have Javascript redirecting users to another site with the same title. Only the redirected site is indexed.

Check for inclusion rules from redirects. This is based on type of redirect. There are three kinds of redirects defined in `EQ$URL`:

- Temporary Redirect: A redirected URL is always allowed if it is a temporary redirection (HTTP status code 302, 307). Temporary redirection is used for whatever reason that the original URL should still be used in the future. It's not possible to find out temporary redirect from `EQ$URL` table other than filtering out the rest from the log file.
- Permanent Redirect: For permanent redirection (HTTP status 301), the redirected URL is subject to boundary rules. Permanent redirection means the original URL is no longer valid and the user should start using the new (redirected) one. In `EQ$URL`, HTTP permanent redirect has the status code 954
- Meta Redirect: Metatag redirection is treated as a permanent redirect. Meta redirect has status code 954. This is always checked against boundary rules.

Check URL Looping

URL looping refers to the scenario where a large number of unique URLs all point to the same document. One particularly difficult situation is where a site contains a large number of pages, and each page contains links to every other page in the site. Ordinarily this would not be a problem, because the crawler eventually analyzes all documents in the site.

However, some Web servers attach parameters to generated URLs to track information across requests. Such Web servers might generate a large number of unique URLs that all point to the same document.

For example, `http://example.com/somedocument.html?p_origin_page=10` might refer to the same document as `http://example.com/somedocument.html?p_origin_page=13` but the `p_origin_page` parameter is different for each link, because the referring pages are different. If a large number of parameters are specified and if the number of referring links is large, then a single unique document could have thousands or tens of thousands of links referring to it. This is an example of how URL looping can occur.

Monitor the crawler statistics in the Oracle SES administration tool to determine which URLs and Web servers are being crawled the most. If you observe an inordinately large number of URL accesses to a particular site or URL, then you might want to do one of the following:

- Exclude the Web Server: This prevents the crawler from crawling any URLs at that host. (You cannot limit the exclusion to a specific port on a host.)
- Reduce the Crawling Depth: This limits the number of levels of referred links the crawler will follow. If you are observing URL looping effects on a particular host, then you should take a visual survey of the site to find out an estimate of the depth of the leaf pages at that site. Leaf pages are pages that do not have any links to other pages. As a general guideline, add three to the leaf page depth, and set the crawling depth to this value.

Be sure to restart the crawler after altering any parameters. Your changes take effect only after restarting the crawler.

What to do Next

If you are still not crawling all the pages you think you should, then check which pages were crawled by doing one of the following:

- Check the crawler log file. (There's a link on the **Home - Schedules** page and the location of the full log on the **Home - Schedules - Status** page.)
- Create a search source group. (**Search - Source Groups - Create New Source Group**) Put only one source in the group. From the **Search** page, search that group. (Click the group name on top of the search box.) Or, from the **Search** page, click **Browse Search Groups**. Click the group name for a hierarchy. You could also click the number next to the group name for a list of the pages crawled.

Tuning Search Performance

This section contains suggestions on how to improve the response time and throughput performance of Oracle SES.

This section contains the most common things to consider to improve search performance:

- Optimize the Index
- Increase the Size of the Indexing Batch Size
- Check the Search Statistics
- Increase the JVM Heap Size
- Increase the Oracle Undo Space

Optimize the Index

Optimizing the **index** reduces fragmentation, and it can significantly increase the speed of searches. Schedule index optimization on a regular basis. Also, optimize the index after the crawler has made substantial updates or if fragmentation is more than 50%. Make sure index optimization is scheduled during off-peak hours. Optimization of a very large index could take several hours.

See the fragmentation level and run index optimization on the **Global Settings - Index Optimization** page in the administration tool.

Increase the Size of the Indexing Batch Size

The data in the cache directory continues to accumulate until it reaches the indexing batch size. When the size is reached, the data is indexed. The bigger the batch size, the less fragmentation in the index. However, the bigger the batch size, the longer it will take to index each batch. Only indexed data can be searched: data in the cache cannot be searched.

Set the indexing batch size on the **Global Settings - Crawler Configuration** page in the administration tool.

Check the Search Statistics

See the **Home - Statistics** page in the administration tool for lists of the most popular queries, failed queries, and ineffective queries. This information can lead to the following actions:

- Refer users to a particular Web site for failed queries on the **Search - Suggested Links** page.
- Fix common errors that users make in searching on the **Search - Alternate Words** page.
- Make important documents easier to find on the **Search - Relevancy Boosting** page.

Relevancy Boosting

Relevancy boosting lets administrators influence the order of documents in the result list for a particular search. You might want to override the default results for the following reasons:

- For a highly popular search, direct users to the best results
- For a search that returns no results, direct users to some results
- For a search that has no click-throughs, direct users to better results

In a search, each result is assigned a score that indicates how relevant the result is to the search; that is, how good a result it is. Sometimes there are documents that you know are highly relevant to some search. For example, your company Web site could have a home page for XML (http://example.com/XML-is-great.htm), which you want to appear high in the results of any search for "XML". You would boost the score of that home page (http://example.com/XML-is-great.htm) to 100 for an "XML" search.

There are two methods for locating URLs for relevancy boosting: *locate by search* or *manual URL entry*.

Note: The document still has a score computed if you enter a search that is not one of the boosted queries.

With relevancy boosting, comparison of the user's query against the boosted queries uses exact string matching. This means that the comparison is case-sensitive and space-aware. Therefore, a document with a boosted score for "Enterprise Search" is not boosted when you enter "search".

Increase the JVM Heap Size

If you expect heavy load on the Oracle SES server, then configure the Java virtual machine (JVM) heap size for better performance.

The heap size is defined in the `$ORACLE_HOME/search/config/searchctl.conf` file. By default, the following values are given:

`max_heap_size` = 1024 megabytes

`min_heap_size` = 512 megabytes

Increase the value of these parameters appropriately. The max size should not exceed the physical memory size. Then restart the mid-tier with `searchctl restart`.

Increase the Oracle Undo Space

Heavy query load should not coincide with heavy crawl activity, especially when there are large-scale changes on the target site. If it does, for example when the crawl needs be scheduled around-the-clock, then increase the size of the Oracle undo tablespace with the `UNDO_RETENTION` parameter.

Using Backup and Recovery

A backup is a copy of configuration data that can be used to recover your configuration settings after a hardware failure. When a backup is performed on the **Global Settings - Configuration Backup and Recovery** page, Oracle SES copies the data to the binary `metaData.bkp` file. The location of that file is provided on the **Global Settings - Configuration Data Backup and Recovery** page. When the backup successfully completes, you must copy this file to a different host. You should backup after making configuration data changes, such as creating or editing sources.

Recovery can only be performed on a fresh installation. When the installation completes, copy the `metaData.bkp` file to the location provided in the administration tool. Sources need to be crawled again to see search results.

Some notes about backup and recovery:

- You must stop all running schedules before doing the backup.
- Secure search does not need to be re-enabled after recovery. If secure search is enabled in the backup instance, you do *not* need to re-register or re-activate the identity plug-in after recovery.

  In 10.1.7, neither re-activation nor re-registration of the identity plug-in is required. If a plug-in was active when the instance was backed up, the same plug-in will be activated in the recovered instance, using the same parameters.

- If you have file or table sources residing on the same machine as the one running Oracle SES, and if you intend to use a different machine for recovery, then you must use the actual host name (not localhost) when creating the sources.
- For database table sources, confirm that the remote tables exist.
- For file sources, confirm that files and paths are valid after recovery.
- During recovery, the mail archive directory settings for existing mailing list and e-mail sources is changed. After recovery, the location will be `<cache-dir>/mail`, which is the default for new e-mail and mailing list sources. Any customized directory locations prior to recovery will be lost.

Integrating with Google Desktop for Enterprise

Oracle Secure Enterprise Search provides a plug-in (or *connector*) to integrate with Google Desktop for Enterprise (GDfE). You can include Google Desktop results in your Oracle SES hitlist. You can also link to Oracle SES from the GDfE interface.

> **See Also:** Google Desktop for Enterprise Readme at `http://host:port/search/query/gdfe/gdfe_readme.html` for details about how to integrate with GDfE

Monitoring Oracle Secure Enterprise Search

In a production environment, where a load balancer or other monitoring tools are used to ensure system availability, Oracle Secure Enterprise Search (SES) can also be easily monitored through the following URL: `http://<host>:<port>/monitor/check.jsp`. The URL should return the following message: **Oracle Enterprise Search instance is up.**

> **Note:** This message is not translated to other languages, because system monitoring tools might need to byte-compare this string.

If Oracle SES is not available, then the URL returns either a connection error or the HTTP status code 503.

Turning On Debug Mode

Debug mode is useful for troubleshooting purposes. To turn on debug mode for Oracle SES administration tool, update the `search.properties` file located in the `$ORACLE_HOME/search/webapp/config` directory. Set `debug=true` and restart the Oracle SES middle tier with `searchctl restart`.

To turn off debug mode when you are finished troubleshooting, set `debug=false` and restart the middle tier with `searchctl restart`.

> **Note:** `$ORACLE_HOME` represents the directory where Oracle SES was installed.
>
> Debug information can be found in the OC4J log file: `$ORACLE_HOME/oc4j/j2ee/OC4J_SEARCH/log/oc4j.log`.

Restarting Oracle Secure Enterprise Search After Rebooting

The tool for starting and stopping the search engine is `searchctl`. To restart Oracle SES (for example, after rebooting the host machine), navigate to the `bin` directory and run `searchctl startall`.

Note: Users are prompted for a password when running `searchctl` commands on UNIX platforms. No password is required on Windows platforms. This is because Oracle SES installation on Windows requires a user with administrator privileges. When running commands to start or stop the search engine, no password is required as long as the user is a member of the administrator group.

See Also: Startup / Shutdown lesson in the Oracle SES admin tutorial:
http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm

6

Oracle Secure Enterprise Search APIs

This chapter explains the Oracle Secure Enterprise Search (SES) APIs and related information. This chapter contains the following topics:

- Overview of Oracle Secure Enterprise Search APIs
- Oracle Secure Enterprise Search Web Services API
- Oracle Secure Enterprise Search SDK

See Also: *Oracle Secure Enterprise Search Java API Reference*

Overview of Oracle Secure Enterprise Search APIs

Oracle Secure Enterprise Search provides the following APIs:

Web Services API

The Web Services API is used to integrate Oracle SES search capabilities into your search application. Oracle SES provides the Web Services Proxy Java library. You can either use the Java library or create proxies, based on the published Web Services Description Language (WSDL), to access Oracle SES Web Services.

Crawler Plug-in API

The Crawler Plug-in API is used to crawl and index proprietary document repositories. This is included in the SDK.

Query-time Authorization API

The Query-time Authorization API filters search results and access to document information at search time. Query-time filtering can be used in addition to, or in place of, ACLs. This is included in the SDK.

URL Rewriter API

The URL Rewriter API is used by the crawler to filter and rewrite extracted URL links before they are inserted into the URL queue. This is included in the SDK.

Oracle Secure Enterprise Search Web Services API

Oracle Secure Enterprise Search Web Services API lets you write your own application to search Oracle SES over the network. The API provides the following benefits:

- Applications can be deployed into any machine that connects to Oracle SES server through a standard Internet protocol.

- Web Services protocol is XML-based, which makes for easy application integration.

Oracle SES also provides the client-side Java proxies for marshalling and parsing Web Services SOAP messages. Client applications can use the library instead of creating SOAP requests and parsing SOAP responses by themselves to access Oracle SES Web Services.

This section contains the following:

Web Services Concepts

Oracle SES Web Services consists of a remote procedure call (RPC) interface to Oracle SES that enables the client application to invoke operations on Oracle SES over the network. The client application uses Web Services Description Language (WSDL) specification published by Oracle SES Web Services URL to send a request message using Simple Object Access Protocol (SOAP). The server then responds to the client application with a SOAP response message.

This section explains the following concepts:

Web Services

A Web Service is a software application identified by a URI whose interfaces and binding are capable of being defined, described, and discovered by XML artifacts. A Web Service supports direct interactions with other software applications using XML-based messages and internet-based products.

A Web Service does the following:

- Exposes and describes itself: A Web Service defines its functionality and attributes so that other applications can understand it. By providing a WSDL file, a Web Service makes its functionality available to other applications.
- Allows other services to locate it on the Web: A Web Service can be registered in a UDDI registry so that applications can locate it.
- Can be invoked: After a Web Service has been located and examined, the remote application can invoke the service using an Internet standard protocol.
- Web Services are of either request and response or one-way style, and they can use either synchronous or asynchronous communication. However, the fundamental

unit of exchange between Web Services clients and Web Services, of either style or type of communication, is a message.

Simple Object Access Protocol

The Simple Object Access Protocol (SOAP) is a lightweight XML-based protocol for exchanging information in a decentralized distributed environment. SOAP supports different styles of information exchange, including RPC-oriented and message-oriented exchange. RPC style information exchange allows for request-response processing, where an endpoint receives a procedure-oriented message and replies with a correlated response message. Message-oriented information exchange supports organizations and applications that need to exchange messages or other types of documents where a message is sent, but the sender might not expect or wait for an immediate response. Message-oriented information exchange is also called document style exchange.

SOAP has the following features:

- Protocol independence
- Language independence
- Platform and operating system independence
- Support for SOAP XML messages incorporating attachments (using the multipart MIME structure)

Web Services Description Language

The Web Services Description Language (WSDL) is an XML format for describing network services containing RPC-oriented and message-oriented information. Programmers or automated development tools can create WSDL files to describe a service and can make the description available over the Internet. Client-side programmers and development tools can use published WSDL specifications to obtain information about available Web Services and to build and create proxies or program templates that access available services.

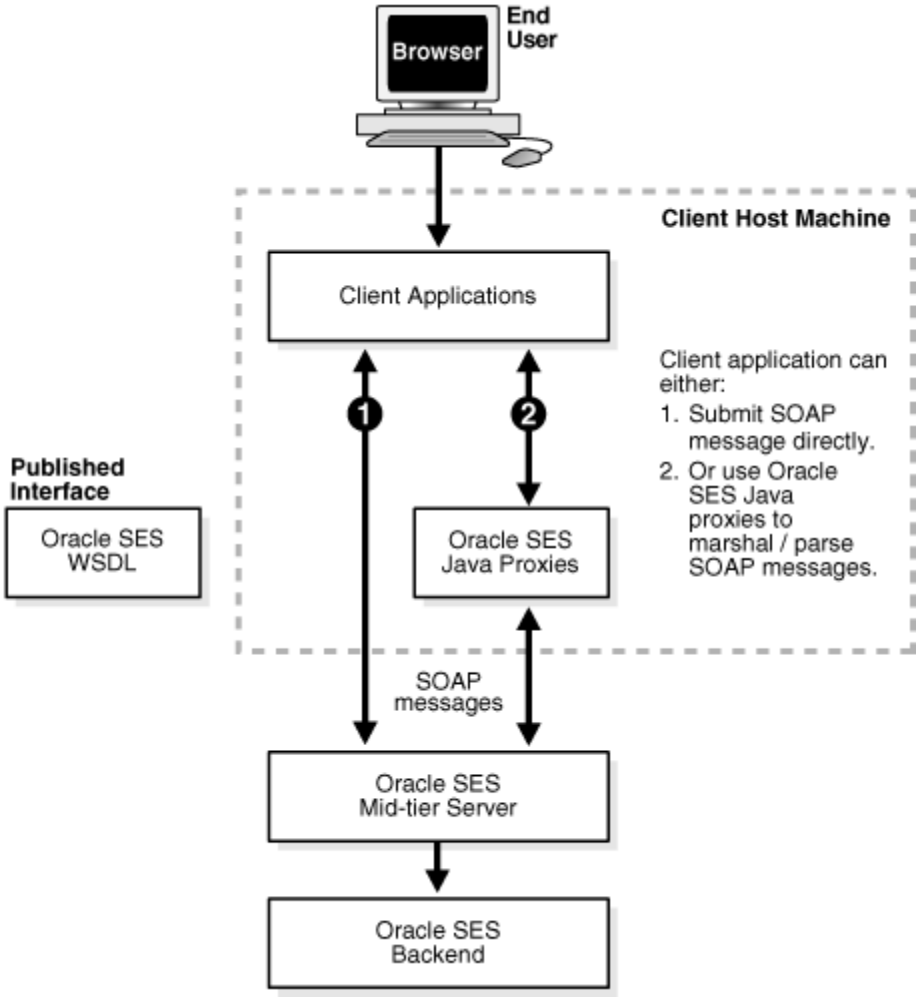
Oracle Secure Enterprise Search Web Services Architecture

Oracle Secure Enterprise Search Web Services is powered by the Oracle SES middle tier OC4J server. The implementation, configuration, and deployment of Oracle SES Web Services follow the procedures and standards provided by OC4J server.

Oracle SES WSDL defines the operations and messages for Oracle SES Web Services. The message exchange of Oracle SES Web Services is RPC style, in which the contents of the SOAP message body conform to a structure that specifies a procedure and includes set of parameters or a response with a result and any additional parameters.

Oracle SES SOAP messages use HTTP binding where a SOAP message is embedded in the body of a HTTP request and a SOAP message is returned in the HTTP response.

The following diagram illustrates the architecture of Oracle SES Web Services:



Development Platforms

You can implement client applications using platforms that support Simple Object Access Protocol (SOAP), such as Oracle JDeveloper, Microsoft .NET, or Apache Axis. These platforms allow you to automatically create code using the Oracle SES WSDL interface. Include the generated code along with the application logic to create a request, invoke the Web Services, and interpret the response.

Oracle Secure Enterprise Search Web Services Operations

Oracle Secure Enterprise Search provides the following categories of Web Services operations:

- Authentication: Authenticate a user's access to Oracle SES. The operation is only required if the user performs secure search.
- Search: Run a search on Oracle SES and obtain a hitlist along with information such as estimated hit count, near duplicate documents in the hitlist, suggested links, and alternate keywords for the executed search.
- Metadata: Obtain the search metadata, such as the list of source groups, the list of supported languages, or the list of search attributes.
- Search Hit: Obtain the search result details, such as the cached version of search result and in-links and out-links of the search hit.

- User Feedback: Send user feedback to Oracle SES, such as user submitted URL.

  See Also: "Oracle Secure Enterprise Search Web Services Operations" on page 6-9

Oracle Secure Enterprise Search Web Services Common Data Types

This section contains the following:

Base Data Types

Oracle Secure Enterprise Search Web Services uses the following base data types:

Table 6–1 Base Data Types

Base Type	Description	Example
xsd:Boolean	Boolean	true, false
xsd:date	Date	2005-12-31
xsd:int	Integer	256
xsd:long	Long integer	12345678900
xsd:string	String	Oracle Secure Enterprise Search

XML-to-Java Data Type Mappings

The mapping between XML schema data types and Java data types depends on the SOAP development environment. The following table shows mappings for the Oracle JDeveloper environment:

Table 6–2 XML-to-Java Type Mappings

XML Schema	Oracle JDeveloper
xsd:Boolean	java.lang.Boolean
xsd:date	java.util.Date
xsd:int	java.lang.Integer
xsd:long	java.lang.Long
xsd:string	java.lang.String

Complex Types

Oracle Secure Enterprise Search Web Services uses the following complex data types:

OracleSearchResult The search result container. It has the following elements:

- `returnCount`: A Boolean value indicating whether the result return count estimate for the hitlist

- `estimatedHitCount`: The estimate count of the search result, -1 means the search result does not return estimated hit count
- `dupRemoved`: A Boolean value indicating whether duplicate documents have been removed from search result
- `dupMarked`: A Boolean value indicating whether duplicate documents have been marked in search result. If `dupRemoved` is true, then `dupMarked` is always false
- `resultElements`: An array of `resultElement`, which represents the actual hitlist
- `suggestedLinks`: An array of `suggestedLink` for the given search
- `query`: The actual search string. The search string should follow Oracle SES query syntax
- `altKeywords`: Alternate keywords (suggestions) for the given search
- `startIndex`: The start index of search results
- `docsReturned`: The number of search hits returned

ResultElement This is the data type for search result element. It has the following elements:

- `author`: Primary author of the document
- `description`: Description of the document
- `url`: URL of the document
- `snippet`: Keywords in context (KWIC) of the document
- `title`: Title of the document
- `lastModified`: Last modified date of the document
- `mimetype`: Mime type of the document
- `score`: Oracle Text score of the document
- `docID`: Document ID
- `language`: Language of the document
- `contentLength`: Content length of the document
- `signature`: Signature of the document
- `infoSourceID`: InfoSource ID of the document
- `infoSourcePath`: InfoSource path of the document
- `groups`: Array of groups to which the document belongs
- `isDuplicate`: Boolean value indicating whether this document is a duplicate of another document in the hitlist
- `hasDuplicate`: Boolean value indicating whether this document has one or more duplicates in the hitlist
- `fedID`: Federated instance ID, used to track which federated instance the document is fetched from
- `customAttributes`: Array of custom nondefault attributes extracted from/for the document during crawling that should be fetched with the results

DataGroup The source group. It has the following elements:

- `groupID`: Source group ID
- `groupName`: Source group name
- `groupDisplayName`: Display name for the source group

Attribute The data type for search attribute. It has the following elements:

- `id`: Search attribute ID
- `name`: Internal name of search attribute
- `displayName`: Display name of search attribute
- `type`: The search attribute type. Value is either 'number', 'string', or 'date'.

Filter The data type for filter condition (predicate). It has the following elements:

- `attributeId`: Search attribute ID
- `attributeType`: Search attribute type. Value is either 'number', 'string', or 'date'.
- `operator`: Operator of the filter condition
  - If the attributeType is 'string', then it should be either 'equals' or 'contains'.
  - If the attributeType is 'number' or 'date', then it should be either 'greaterthan', 'lessthan', or 'equals'.
- `attributeValue`: value of the filter condition (predicate)
  - For 'string' type attribute, the value is simply the string itself.
  - For 'number' type attribute, the value should be represented by a string consisting of an optional sign, '+' or '-', followed by a sequence of zero or more decimal digits ("the integer"), optionally followed by a fraction. The fraction consists of a decimal point followed by zero or more decimal digits. The string must contain at least one digit in either the integer or the fraction.
  - For 'date' type attribute, the value should be in the format 'mm/dd/yyyy', where mm is the month (00~12), dd is the date (01~31), yyyy is the year (for example, 2005)

Examples:

- If the filter condition is Title contains 'Oracle SES', then the client application needs to lookup the attribute ID of search attribute 'Title' and include the following (element, value) pairs:
  - attributeID => 1 (assuming the search attribute id of 'Title' is 1)
  - operator => contains
  - attributeValue => Oracle Secure Enterprise Search
- If the filter condition is Price greater than 1000, then the client application needs to lookup the attribute ID of search attribute 'Price' and include the following (element, value) pairs:
  - attributeID => 2 (assuming the search attribute id of 'Price' is 2)
  - operator => greaterthan
  - attributeValue => 1000

Node This is the data type for the infosource node. It has the following elements:

- `id`: Infosource node ID
- `fedId`: Federated instance ID, used to track which federated instance the node belongs to
- `name`: Name of the node
- `docCount`: Number of documents under the node. If the value is –1, then there exists documents under the node but the count cannot be shown.
- `hasChildren`: Indicates if the node has any children
- `fullpath`: Full path of the category node
- `fullpathIds`: The IDs of each node in the full path

AttributeLOVElement This is the element of 'AttributeLOV', the list of search attribute values. It has the following elements:

- `value`: Attribute value (internal value)
- `displayValue`: Display value

SessionContextElement This data structure is used to store authentication information for the search user in the form of a name-value pair, which can be used during query-time authorization filtering of the results. It has following elements:

- `name`: Name of the authentication attribute
- `value`: Value of the authentication attribute

Status This is the status of the request. It has the following elements:

- `status`: Status code. Value is either 'successful' or 'failed'
- `message`: Status message. Value is null, or an error message if the status is 'failed'

Language This is the language data type. It has the following elements:

- `languageName`: Name of the language
- `languageDisplayName`: Display name (translated name) of the language

CustomAttributes

This data type stores the custom (nondefault) attributes extracted for a document during crawling. It has following elements:

- `name`: Custom-attribute name
- `value`: Custom-attribute value

Array Types

Oracle Secure Enterprise Search Web Services uses the following complex array types:

- `AttributeArray`: Array of Attribute
- `AttributeLOVElementArray`: Array of AttributeLOVElement
- `CustomAttributeArray`: Array of CustomAttribute
- `DataGroupArray`: Array of DataGroup
- `FilterArray`: Array of Filter

- `IntArray`: Array of int
- `LanguageArray`: Array of Language
- `NodeArray`: Array of Node
- `ResultElementArray`: Array of ResultElement
- `SessionContextElementArray`: Array of SessionContextElement
- `StringArray`: Array of String

See Also: Appendix C, "WSDL Specification"

Oracle Secure Enterprise Search Web Services Operations

This section contains the following:

- Authentication Operations
- Search Operations
- Metadata Operations
- Search Hit Operations
- User Feedback Operations

Authentication Operations

This section describes the following authentication operations:

- loginRequest Message
- loginResponse Message
- logoutRequest Message
- logoutResponse Message
- setSessionContextRequest Message
- setSessionContextResponse Message

loginRequest Message This message requests Oracle SES to authenticate the search user. It consists of the following parameters:

- `username`: User name for the search user
- `password`: Password for the search user

```
<message name="loginRequest">
   <part name="username"      type="xsd:string"/>
   <part name="password"      type="xsd:string"/>
</message>
```

loginResponse Message This message contains the return status for the `loginRequest` message.

```
<message name="loginResponse">
   <part name="return"      type="typens:Status"/>
</message>
```

logoutRequest Message This message is used when the user logs out from the search application.

```
<message name="logoutRequest">
</message>
```

logoutResponse Message This message contains of the return status for the `logoutRequest` message.

```
<message name="logoutResponse">
   <part name="return"      type="typens:Status"/>
</message>
```

setSessionContextRequest Message This message is used to pass authentication information for the search user, which can be used during query-time filtering.

> **Note:** Login and logout Web Services calls cause Oracle SES to automatically set or reset the `AUTH_USER` value in the session context that is passed to the query-time filter. This session context attribute cannot be overwritten explicitly through the `setSessionContext` call.

It consists of following parameter:

- `sessionContext`: An array of `SessionContextElement`. This array stores the authentication information needed for the query-time authentication filtering in the form of name-value pairs.

```
<message name="setSessionContextRequest">
    <part name="sessionContext"     type="typens:SessionContextElementArray"/>
</message>
```

setSessionContextResponse Message This message contains the return status for the `setSessionContext` message.

```
<message name="setSessionContextResponse">
  <part name="return"             type="typens:Status"/>
</message>
```

Search Operations

This section describes the following search operations:

- doOracleSearch Message
- doOracleSearchResponse Message
- doOracleBrowseSearch Message
- doOracleBrowseSearchResponse Message
- doOracleSimpleSearch Message
- doOracleSimpleSearchResponse Message

doOracleSearch Message This is the main message for the search application. It consists of the following parameters:

- `query`: A search string. It must be a valid string and it cannot be null. The search string should follow Oracle SES query syntax. See "Oracle Secure Enterprise Search Web Services Query Syntax" on page 6-19 for details.
- `startIndex`: The index of the first result to be returned. For example, if there are 67 results, you might want to start at 20. The default is 1 if not set explicitly.

- `docsRequested`: The maximum number of results to be returned. The default is 10 if not set explicitly.
- `dupRemoved`: Enable or disable duplicate removal. If turned on, the search result will eliminate all duplicate and near duplicate documents from the result list. The `dupMarked` switch will have no effect when `dupRemoved` is turned on. The default is false if not set explicitly.
- `dupMarked`: Enable or disable duplicate detection. If `dupRemoved` is turned off and `dupMarked` is turned on, then the search result will keep all duplicate and near duplicate documents from the result list and mark them as duplicates. If `dupRemoved` is turned on, then the `dupMarked` switch will have no effect. The default is false if not set explicitly.
- `groups`: Limit the search result to the documents from specified source groups. The default is for all groups if not set explicitly.
- `queryLang`: Set the language of search. This is equivalent to `locale`. For relevancy boosting to work, `queryLang` must be set.
- `docLang`: Set the document language to limit the search to documents of a particular language. The default is English ("en") if not set explicitly.
- `returnCount`: Set to true to return total hit count with the result. The default is false if not set explicitly.
- `filterConnector`: The connector between all filters: "and" indicates the search result must satisfy all filters, "or" indicates the search result just needs to satisfy at least one filter. The default is "and" if not set explicitly.
- `filters`: An array of filters. Each filter is a restriction on search results. Filters are connected by `filterConnector`. The default is null (no filter applies to the search result) if not set explicitly.
- `fetchAttributes`: Array of integers representing the nondefault attribute IDs to be fetched in the resultElements. The default is null (no attributes other than default-attributes are fetched in the resultElements).

```
<message name="doOracleSearch">
   <part name="query"            type="xsd:string"/>
   <part name="startIndex"       type="xsd:int"/>
   <part name="docsRequested"    type="xsd:int"/>
   <part name="dupRemoved"       type="xsd:boolean"/>
   <part name="dupMarked"        type="xsd:boolean"/>
   <part name="groups"           type="typens:DataGroupArray"/>
   <part name="queryLang"        type="xsd:string"/>
   <part name="docLang"          type="xsd:string"/>
   <part name="returnCount"      type="xsd:boolean"/>
   <part name="filterConnector"  type="xsd:string"/>
   <part name="filters"          type="typens:FilterArray"/>
   <part name="fetchAttributes"  type="typens:IntArray"/>
</message>
```

doOracleSearchResponse Message This message returns the search result in `OracleSearchResult` data type.

```
<message name="doOracleSearchResponse">
    <part name="return"  type="typens:OracleSearchResult"/>
</message>
```

doOracleBrowseSearch Message This message restricts a search to a particular node. It consists of the following parameters:

- `query`: A search string. It must be a valid string, and it cannot be null. The search string should follow Oracle SES query syntax. See "Oracle Secure Enterprise Search Web Services Query Syntax" on page 6-19 for more details.
- `nodeID`: The ID of the node to restrict the search to.
- `fedID`: The ID of the federated instance the parent node belongs to ("-1" for local node).
- `startIndex`: The index of the first result to be returned. For example, if there are 67 results, then you might want to start at 20. The default is 1 if not set explicitly.
- `docsRequested`: The maximum number of results to be returned. The default is 10 if not set explicitly.
- `dupRemoved`: Enable or disable duplicate removal. If turned on, then the search result will eliminate all duplicate and near duplicate documents from the result list, and the `dupMarked` switch will have no effect when `dupRemoved` is turned on. The default is false if not set explicitly.
- `dupMarked`: Enable or disable duplicate detection. If `dupRemoved` is turned off and `dupMarked` is turned on, then the search result will keep all duplicate and near duplicate documents from the result list and mark them as duplicates. If `dupRemoved` is turned on, then the `dupMarked` switch will have no effect. The default is false if not set explicitly.
- `queryLang`: Set the language to limit the search to documents of a particular language. This is equivalent to `locale`. For relevancy boosting to work, `queryLang` must be set.
- `docLang`: Set the document language to limit the search. If the value is not specified, then it will return documents in all languages.
- `returnCount`: Set to true to return total hit count with the result. The default is false if not set explicitly.
- `fetchAttributes`: Array of integers representing the nondefault attribute IDs to be fetched in the `resultElements`. The default is null (no attributes other than default attributes are fetched in the `resultElements`).

```
<message name="doOracleBrowseSearch">
  <part name="query"              type="xsd:string"/>
  <part name="nodeID"             type="xsd:string"/>
  <part name="fedID"              type="xsd:string"/>
  <part name="startIndex"         type="xsd:int"/>
  <part name="docsRequested"      type="xsd:int"/>
  <part name="dupRemoved"         type="xsd:boolean"/>
  <part name="dupMarked"          type="xsd:boolean"/>
  <part name="queryLang"          type="xsd:string"/>
  <part name="docLang"            type="xsd:string"/>
  <part name="returnCount"        type="xsd:boolean"/>
  <part name="fetchAttributes"    type="typens:IntArray"/>
</message>
```

doOracleBrowseSearchResponse Message This message returns the search result in `OracleSearchResult` data type.

```
<message name="doOracleBrowseSearchResponse">
    <part name="return"  type="typens:OracleSearchResult"/>
</message>
```

doOracleSimpleSearch Message This is a simplified form of the doOracleSearch message. In this message you don't need to specify the advanced search parameters that are specified in the doOracleSearch message. It consists of following parameters:

- `query`: A search string. It must be a valid string and it cannot be null. The search string should follow Oracle SES query syntax. See "Oracle Secure Enterprise Search Web Services Query Syntax" on page 6-19 for details.
- `startIndex`: The index of the first result to be returned. For example, if there are 67 results, you might want to start at 20. The default is 1, if not set explicitly.
- `docsRequested`: The maximum number of results to be returned. The default is 10, if not set explicitly.
- `dupRemoved`: Enable or disable duplicate removal. If turned on, then the search result will eliminate all duplicate and near duplicate documents from the result list. The `dupMarked` switch will have no effect when `dupRemoved` is turned on. The default is false if not set explicitly.
- `dupMarked`: Enable or disable duplicate detection. If `dupRemoved` is turned off and `dupMarked` is turned on, then the search result will keep all duplicate and near duplicate documents from the result list and mark them as duplicates. If `dupRemoved` is turned on, then the `dupMarked` switch will have no effect. The default is false if not set explicitly.
- `returnCount`: Set to true to return total hit count with the result. The default is false if not set explicitly.

```
<message name="doOracleSimpleSearch">
  <part name="query"              type="xsd:string"/>
  <part name="startIndex"         type="xsd:int"/>
  <part name="docsRequested"      type="xsd:int"/>
  <part name="dupRemoved"         type="xsd:boolean"/>
  <part name="dupMarked"          type="xsd:boolean"/>
  <part name="returnCount"        type="xsd:boolean"/>
</message>
```

doOracleSimpleSearchResponse Message This message returns the search result in `OracleSearchResult` data type.

```
<message name="doOracleSimpleSearchResponse">
  <part name="return"         type="typens:OracleSearchResult"/>
</message>
```

Browse Operations

This section describes the following browse operations:

- getInfoSourceNodesRequest Message
- getInfoSourceNodesResponse Message
- getInfoSourceAncestorNodesRequest Message
- getInfoSourceAncestorNodesResponse Message
- getInfoSourceNodeRequest Message
- getInfoSourceNodeResponse Message

getInfoSourceNodesRequest Message This message gets the list of info source nodes given the parent node ID. It consists of the following parameters:

- `parentNodeID`: The node ID for which all children nodes will be returned. If it is not set, then the message will return all the root nodes.
- `fedID`: The ID of the federated instance the parent node belongs to ("-1" for local node).
- `locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getInfoSourceNodesRequest">
    <part name="parentNodeID"      type="xsd:string"/>
    <part name="fedID"             type="xsd:string"/>
    <part name="locale"            type="xsd:string"/>
</message>
```

getInfoSourceNodesResponse Message This message returns an array of info source nodes.

```
<message name="getInfoSourceNodesResponse">
    <part name="nodes"    type="typens:NodeArray"/>
</message>
```

getInfoSourceAncestorNodesRequest Message This message gets the full path of a node, from root to node, given an info source node. It consists of the following parameters:

- `nodeID`: The node ID for which all the nodes in the path from root to node will be returned, nodeID must be set and it cannot be null.
- `locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getInfoSourceAncestorNodesRequest">
    <part name="nodeID"        type="xsd:string"/>
    <part name="locale"        type="xsd:string"/>
</message>
```

getInfoSourceAncestorNodesResponse Message This message returns an array of info source ancestor nodes.

```
<message name="getInfoSourceAncestorNodesResponse">
    <part name="nodes"    type="typens:NodeArray"/>
</message>
```

getInfoSourceNodeRequest Message This message retrieves a particular node. It consists of the following parameters:

- `nodeID`: The node ID of the node to get, nodeID must be set and it cannot be null.
- `fedID`: The ID of the federated instance the parent node belongs to ("-1" for local node).
- `locale`: A two letter representation of Locale, the default is English ("en") if not set explicitly.

Message format:

```
<message name="getInfoSourceNodeRequest">
   <part name="nodeID        "type="xsd:string"/>
   <part name="fedID"        "type="xsd:string"/>
   <part name="locale        "type="xsd:string"/>
</message>
```

getInfoSourceNodeResponse Message This message returns the node requested.

```
<message name="getInfoSourceNodeResponse">
    <part name="node    "type="typens:Node"/>
</message>
```

Metadata Operations

This section describes the following metadata operations:

- getLanguageRequest Message
- getLanguageResponse Message
- getDataGroupsRequest Message
- getDataGroupsResponse Message
- getAttributesRequest Message
- getAttributesResponse Message
- getAllAttributesRequest Message
- getAllAttributesResponse Message
- getAttributeLOVRequest Message
- getAttributeLOVResponse Message

getLanguageRequest Message This message gets all the languages supported by Oracle SES. It is used by the client application to display the list of languages. It consists of the following parameter:

`locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getLanguagesRequest">
    <part name="locale"        type="xsd:string"/>
</message>
```

getLanguageResponse Message

This message returns all supported languages.

```
<message name="getLanguagesResponse">
    <part name="return"        type="typens:LanguageArray"/>
</message>
```

getDataGroupsRequest Message This message requests for all source groups defined in Oracle SES. It is used by the client application to show all source groups in the search page, such that the end user can restrict their search results within one or multiple source groups. It consists of the following parameter:

`locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getDataGroupsRequest">
    <part name="locale"        type="xsd:string"/>
</message>
```

getDataGroupsResponse Message This message returns all source groups defined in Oracle SES.

```
<message name="getDataGroupsResponse">
    <part name="groups"        type="typens:DataGroupArray"/>
</message>
```

getAttributesRequest Message This message gets a list of search attributes that applied to the given source groups. It consists of the following parameters:

- `locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.
- `groups`: Limit the request to the attributes from specified source groups. The default is all groups if not set explicitly.
- `groupConnector`: The connector between all groups: "and" indicates the response is the attributes available in the set of source groups by finding the intersection of each group's attributes, "or" indicates the response is the attributes available in the set of source groups by finding the union of each group's attributes. The default is "or" if not set explicitly.

```
<message name="getAttributesRequest">
    <part name="locale"          type="xsd:string"/>
    <part name="groups"          type="typens:DataGroupArray"/>
    <part name="groupConnector"  type="xsd:string"/>
</message>
```

getAttributesResponse Message This message returns an array of search attributes.

```
<message name="getAttributesResponse">
    <part name="return"       type="typens:AttributeArray"/>
</message>
```

getAllAttributesRequest Message This message gets all search attributes defined in Oracle SES. It consists of the following parameter:

`locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getAllAttributesRequest">
  <part name="locale" type="xsd:string"/>
</message>
```

getAllAttributesResponse Message This message returns all search attributes defined in Oracle SES.

```
<message name="getAllAttributesResponse">
    <part name="return"       type="typens:AttributeArray"/>
</message>
```

getAttributeLOVRequest Message This message gets the LOV items given a search attribute. It consists of the following parameters:

- `attribute`: A search attribute for the LOV (list of values) requested.
- `locale`: A two letter representation of locale. The default is English ("en") if not set explicitly.

```
<message name="getAttributeLOVRequest">
  <part name="attribute"     type="typens:Attribute"/>
  <part name="locale"        type="xsd:string"/>
</message>
```

getAttributeLOVResponse Message This message returns an array of search attribute LOV elements.

```
<message name="getAttributeLOVResponse">
  <part name="return"    type="typens:AttributeLOVElementArray"/>
</message>
```

Search Hit Operations

This section describes the following search hit operations:

- getCachedPageRequest Message
- getCachedPageResponse Message
- getInLinksRequest Message
- getInLinksResponse Message
- getOutLinksRequest Message
- getOutLinksResponse Message
- logUserClickRequest Message
- logUserClickResponse Message

getCachedPageRequest Message This message gets the cached version of a document given the document ID and the search string. The search string will be highlighted in the output. It consists of the following parameters:

- `query`: The search string
- `docID`: The document ID to be fetched
- `fedID`: The federated instance ID, used to track which federated instance the document is fetched from

```
<message name="getCachedPageRequest">
    <part name="query"          type="xsd:string"/>
    <part name="docID"          type="xsd:int"/>
    <part name="fedID"          type="xsd:string"/>
</message>
```

getCachedPageResponse Message This message returns the byte array of the cached HTML page.

```
<message name="getCachedPageResponse">
    <part name="return"         type="xsd:base64Binary"/>
</message>
```

getInLinksRequest Message This message gets all the incoming links for a given search hit (document). It consists of the following parameters:

- `docID`: The document ID for which the incoming links to be fetched. It must be a valid document ID and it cannot be null.
- `maxNum`: The maximum number of incoming links requested. The default is 25 if not set explicitly.
- `fedID`: The federated instance ID, used to track which federated instance the document is fetched from

```
<message name="getInLinksRequest">
    <part name="docID"                type="xsd:int"/>
    <part name="maxNum"               type="xsd:int"/>
    <part name="fedID"                type="xsd:string"/>
</message>
```

getInLinksResponse Message This message returns an array of incoming link URL strings.

```
<message name="getInLinksResponse">
    <part name="return"      type="typens:StringArray"/>
</message>
```

getOutLinksRequest Message This message gets all the outgoing links for a given search hit (document). It consists of the following parameters:

- `docID`: The document ID for which the outgoing links to be fetched. It must be a valid document ID and it cannot be null.
- `maxNum`: The maximum number of outgoing links requested. The default is 25 if not set explicitly.
- `fedID`: The federated instance ID, used to track which federated instance the document is fetched from

```
<message name="getOutLinksRequest">
    <part name="docID"                type="xsd:int"/>
    <part name="maxNum"               type="xsd:int"/>
    <part name="fedID"                type="xsd:string"/>
</message>
```

getOutLinksResponse Message This message returns an array of outgoing link URL strings.

```
<message name="getOutLinksResponse">
    <part name="return"      type="typens:StringArray"/>
</message>
```

logUserClickRequest Message This message logs the user's click. It consists of the following parameters:

- `queryID`: ID of the submitted search
- `urlID`: ID of the document that the user clicked on
- `infosourceID`: Infosource ID. If none, then –1 is used as the default value
- `position`: The position of the document in the result list (for example, first hit on the page or 9th hit on the page)
- `fedID`: Federation ID. Specifies the federated instance on which the document resides.

```
<message name="logUserClickRequest">
    <part name="queryID"        type="xsd:int"/>
    <part name="urlID"          type="xsd:int"/>
    <part name="infoSourceID"   type="xsd:int"/>
    <part name="position"       type="xsd:int"/>
    <part name="fedID"          type="xsd:string"/>
</message>
```

logUserClickResponse Message This message returns the URL of the clicked-on document.

```
<message name="logUserClickResponse">
  <part name="url"              type="xsd:string"/>
</message>
```

User Feedback Operations

This section describes the following user feedback operations:

- submitUrlRequest Message
- submitUrlResponse Message

submitUrlRequest Message This message submits a URL to Oracle SES, such that Oracle SES will crawl and index the URL. It consists of the following parameter:

`url`: The URL to be submitted to the crawler so it can be crawled next time. It must be a valid URL and it cannot be null.

```
<message name="submitUrlRequest">
     <part name="url"        type="xsd:string"/>
</message>
```

submitUrlResponse Message This message returns the status, which consists of two strings: the first one is the submission status, it is either "successful" or "failed"; the second string is the error message in case that submission status is "failed".

```
<message name="submitUrlResponse">
     <part name="return"      type="typens:Status"/>
</message>
```

Oracle Secure Enterprise Search Web Services Query Syntax

This section describes the query syntax used in the Oracle Secure Enterprise Search Search API.

Search Term

A search term can be a single word, a phrase, or a special search term. For example, if the search string is oracle secure enterprise search, then there are four search terms within the search string: oracle, secure, enterprise, and search. If the search string is oracle "secure enterprise search", then there are two search terms within the search string: oracle and "secure enterprise search".

Search terms in different cases are treated the same (case insensitive). For example, searching oracle, Oracle, or ORACLE will return the same search result.

Phrase

A phrase is a string enclosed in double-quotes ("). It can contain one or multiple words.

Operators

The following operators are defined in the query syntax:

- Plus [+]: The plus operator specifies that the search term immediately following it must be found in all matching documents. For example, searching for [Oracle +Applications] only finds documents that contain the word "Applications". In a multiple word search, you can attach a [+] in front of every token including the very first token. A token is a phrase enclosed in double-quotes ("). It can be a

single word or a phrase, but there should be no space between the [+] and the token.

- Minus [-]: The minus operator specifies that the search term immediately following it cannot appear in any document included in the search result. For example, searching for [Oracle -Applications] only finds documents that do not contain the word "Applications". In a multiple word search, you can attach a [-] in front of every token except the very first token. It can be a single word or a phrase, but there should be no space between the [-] and the token.
- Asterisk [*]: The asterisk specifies a wildcard search. For example, searching for the string [Ora*] finds documents that contain all words beginning with "Ora" such as "Oracle" and "Orator". You can also insert an asterisk in the middle of a word. For example, searching for the string [A*e] finds documents that contain words such as "Apple" or "Ape".

Default Search - Implicit AND Search

By default, Oracle SES searches all of your search terms, as well as relevant variations of the terms you have entered. There is no need to include any operators (like 'AND') between terms. The order of the terms in the search will affect the search results.

Word Separator

Use one or more space characters ' ' to separate each of the search terms.

Filter Conditions (Advanced Conditions)

Oracle SES query syntax only supports 'Site' and 'File type' filter conditions. It does not support any other filter conditions (advanced conditions) such as title, author, last modified date. To restrict your search with other filter conditions, you can specify them in the Web Services API message `doOracleSearch`.

Special Search Terms

Oracle SES supports the use of several special search terms that allow the user or search administrator to access additional capabilities of the Oracle SES in front of it. Following is the list of special search terms:

'Exclude' Search Term You can exclude a word from your search by putting a minus sign [-] immediately in front of the term you want to exclude from the search results. Exclusion does not work with stop words.

Example: oracle -search

Negative search is not allowed unless there is another positive search term. For example:

`-search` is an invalid search.

`oracle -search` is a valid search.

Wildcard Search Search for words starting with "ora". The asterisk can only be specified at the end (right side) or middle of a search term. So you cannot search for something like *earch.

Example: Ora*

Phrase Search Search for complete phrases by enclosing them in quotation marks. Words marked in this way will appear together in all results exactly as entered.

Example: "oracle secure enterprise search"

Site Restricted Search If you know the specific Web site you want to search, but are not sure where the information is located within that site, then search only within the specific Web site. Enter the search followed by the string "site:" followed by the host name.

Example: oracle site:text.us.oracle.com

Notes:

- Domain restriction is not supported, because Oracle SES does not support left-truncated wildcard search (such as *.oracle.com)
- The exclusion operator (-) can be applied to this search term to remove a Web site from consideration in the search.
- Site restricted search term is implicit AND with other search terms.
- Only one site restriction is allowed. Also, you cannot have both site inclusion and exclusion in the search string. For example, the following search string is invalid:

```
oracle search  site:www.oracle.com  -site:otn.oracle.com
```

File Type Restricted Search The search prefix "filetype:" filters the results returned to include only documents with the extension specified immediately after. There can be no space between "filetype:"; and the specified extension.

Example: oracle filetype:doc

Notes:

- The exclusion operator (-) can be applied to this search term to remove a file type from consideration in the search.
- Only one file type can be included. The following extensions are supported: doc, html, pdf, txt, rtf, ppt, ps, and xls.
- File type restricted search term is implicit AND with other search terms.
- Only one file type restriction is allowed. Also, you cannot have both file type inclusion and exclusion in the search string. For example, the following search string is invalid:

```
oracle search  filetype:doc -filetype:pdf
```

Oracle Secure Enterprise Search Web Services Example

Following is a simple JSP application using Oracle Secure Enterprise Search proxy Java library to provide the basic search functionality:

```
<%@page contentType="text/html; charset=utf-8" %>
<%@page import = "java.util.Vector" %>
<%@page import = "java.net.URL" %>
<%@page import = "java.util.Properties" %>
<%@page import = "java.util.HashMap" %>
<%@page import = "org.apache.soap.Header" %>
<%@page import = "org.apache.soap.rpc.Call" %>
<%@page import = "org.apache.soap.rpc.Parameter" %>
<%@page import = "org.apache.soap.rpc.Response" %>
<%@page import = "org.apache.soap.Fault" %>
<%@page import = "org.apache.soap.SOAPException" %>
<%@page import = "org.apache.soap.Constants" %>
<%@page import = "org.apache.soap.encoding.SOAPMappingRegistry" %>
```

```
<%@page import = "org.apache.soap.encoding.soapenc.BeanSerializer" %>
<%@page import = "org.apache.soap.util.xml.QName" %>
<%@page import = "oracle.soap.transport.http.OracleSOAPHTTPConnection" %>
<%@page import = "oracle.soap.encoding.soapenc.EncUtils" %>
<%@page import = "oracle.search.query.webservice.client.*" %>

<%
  //
  // Get the search term entered by the user
  //
  String searchTerm = request.getParameter("searchTerm");
  if (searchTerm == null)  searchTerm = "";

  //
  // Define the result element array.
  //
  ResultElement[] resElemArray = null; // ResultElement is one of the proxy Java
classes
  int estimatedHitCount = 0;

  if (searchTerm != null && !"".equals(searchTerm))
  {
    //
    // Create the Oracle SES Web Services client stub
    //
    OracleSearchService stub = new OracleSearchService();

    //
    // Set the Oracle SES Web Services URL.
    // The URL is http://<host>:<port>/search/query/OracleSearch
    //
    stub.setSoapURL("http://staca19:7777/search/query/OracleSearch");

    //
    // Get the search result by calling OracleSearchService.doOracleSearch()
    //
    OracleSearchResult result = stub.doOracleSearch(searchTerm,
                            new Integer(1),
                            new Integer(10),
                            Boolean.TRUE,
                            Boolean.TRUE,
                            null,
                            "en",
                            "en",
                            Boolean.TRUE,
                            null,
                            null,
                            null);
    //
    // Get the estimated hit count by calling
    estimatedHitCount = result.getEstimatedHitCount().intValue();

    // Get the search results
    resElemArray = result.getResultElements();
  }
%>

<HTML>
<HEAD>
    <TITLE>Oracle SES Web Services Demo </TITLE>
```

```
</HEAD>
<BODY>
<FORM name="searchBox" method="post" action="./DemoWS.jsp">
 <INPUT id="inputMain" type="text" size="40" name="searchTerm"
value="<%=searchTerm%>">
 <INPUT type="hidden"  name="searchTerm" value="<%= searchTerm %>">
 <INPUT type="submit" name="action" value="Search">
</FORM>
<BR><BR><BR>

<%
  //
  // Render the search results
  //
  if (resElemArray == null || resElemArray.length == 0)
  {
%>
  <H3> There are no matches for the search term </H3>
<%
  }
  else
  {
%>
  <H3> There are about <%=estimatedHitCount%> matches </H3>
<%
    for (int i=0; i<resElemArray.length; i++)
    {
      String title = resElemArray[i].getTitle();
      if (title == null) title = "Untitled Document";
%>
  <P>
    <B><A HREF="<%=resElemArray[i].getUrl()%>"><%=title%></A> </B>
    <BR>
     <%=resElemArray[i].getSnippet()%>
    <BR>
  </P>
<%
    }
  }
%>
</BODY>
</HTML>
```

Oracle Secure Enterprise Search Web Services Installation

Oracle SES Web Services runs on top of Oracle SES middle tier standalone OC4J server. It is installed and configured as part of the default install option. You can use Oracle SES Web Services out-of-the-box. There is no specific step to administrate Oracle SES Web Services. Follow the same middle tier administration steps to start and stop Oracle SES Web Services.

Your search application needs to access the following Oracle SES Web Services URL:

`http://<server_name>:<port_number>/search/query/OracleSearch`

For example, if your Oracle SES middle tier is running on host 'myhost' and the port number is 8888, then the Web Services URL is the following:

`http://myhost:8888/search/query/OracleSearch`

There is a default Oracle SES Web Services administrator console provided by OC4J. The administrator console URL is the same as the Oracle SES Web Services URL. You can obtain the following information from the administrator console:

- Oracle SES WSDL description
- List of Web Services messages and operations
- Client-side Java proxies and source codes

Client-Side Java Proxy Library

Oracle SES also provides client-side Java proxies for marshalling and parsing Web Services SOAP messages. Client applications can use the library to access Oracle SES Web Services.

The proxy library includes the following Java classes, which are mapped to the corresponding Web Services data types and messages:

- `oracle.search.search.webservice.client.Attribute`
- `oracle.search.search.webservice.client.AttributeLOVElement`
- `oracle.search.search.webservice.client.CustomAttribute`
- `oracle.search.search.webservice.client.DataGroup`
- `oracle.search.search.webservice.client.Filter`
- `oracle.search.search.webservice.client.Language`
- `oracle.search.search.webservice.client.Node`
- `oracle.search.search.webservice.client.OracleSearchResult`
- `oracle.search.search.webservice.client.OracleSearchService`
- `oracle.search.search.webservice.client.ResultElement`
- `oracle.search.search.webservice.client.SessionContextElement`
- `oracle.search.search.webservice.client.Status`
- `oracle.search.search.webservice.client.SuggestedLink`

To compile and run your client application using the Oracle SES client-side Java proxy library, you need to include the following files in the Java CLASSPATH. You can obtain these files from Oracle SES server file directory.

- `$ORACLE_HOME/search/lib/search_query.jar` (The proxy Java library)
- `$ORACLE_HOME/oc4j/webservices/lib/soap.jar`
- `$ORACLE_HOME/oc4j/j2ee/home/lib/http_client.jar`
- `$ORACLE_HOME/lib/xmlparserv2.jar`
- `$ORACLE_HOME/lib/mail.jar`
- `$ORACLE_HOME/lib/activation.jar`

Internally Used Web Services Messages

The following Web Services messages and operations are intended for Oracle SES internal use only. ***They are subject to change or removal in future releases.***

- `setSearchUserRequest`, `setSearchUserResponse`, `setSearchUser`
- `proxyLoginRequest`, `proxyLoginResponse`, `proxyLogin`

Oracle Secure Enterprise Search SDK

The Oracle Secure Enterprise Search SDK contains the following APIs:

- Crawler Plug-in API
- URL Rewriter API
- Query-time Authorization API

Crawler Plug-in API

You can implement a crawler plug-in to crawl and index a proprietary document repository. In Oracle SES, the proprietary repository is called a *user-defined source*. The module that enables the crawler to access the source is called a crawler plug-in (or *connector*).

The plug-in collects document URLs and associated metadata from the user-defined source and returns the information to the Oracle SES crawler. The crawler starts processing each URL as it is collected. The crawler plug-in must be implemented in Java using the Oracle SES Crawler Plug-in API. Crawler plug-ins go in the `$ORACLE_HOME/search/lib/plugins` directory.

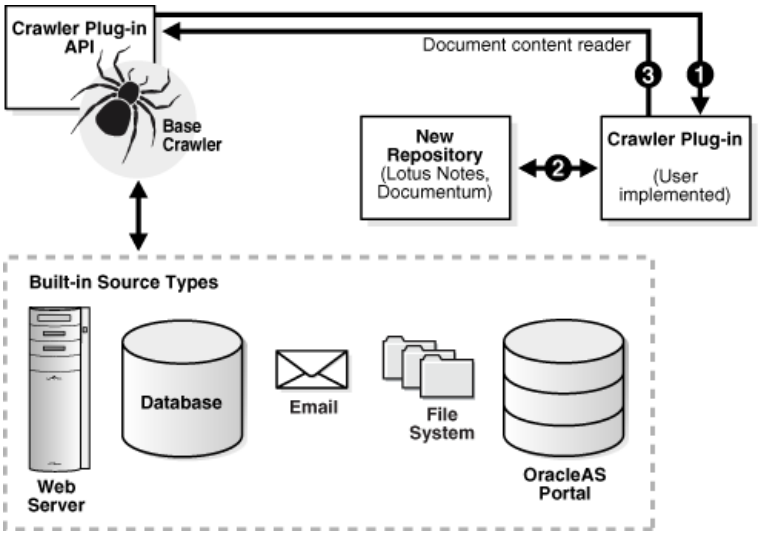
This section includes the following topics:

- Crawler Plug-in Overview
- Crawler Plug-in Functionality

> **See Also:** Oracle SES developer tutorial for a guide to using the Crawler Plug-in API:
>
> `http://st-curriculum.oracle.com/tutorial/SESDevTutorial/index.htm`

Crawler Plug-in Overview

The following diagram illustrates the crawler plug-in architecture.



Two interfaces in the Crawler Plug-in API (`CrawlerPluginManager` and `CrawlerPlugin`) need to be implemented to create a crawler plug-in. A crawler plug-in does the following:

- Provides the metadata of the document in the form of document attributes
- Provides access control list information (ACL) if the document is protected.
- Maps each document attribute to a common attribute name used by end users
- Optionally provides the list of URLs that have changed since a given time stamp
- Optionally provides an access URL in addition to the display URL for the processing of the document
- Provide the document contents in the form of a Java Reader. In other words, the plug-in is responsible for fetching the document.
- Can submit "attribute-only" documents to the crawler; that is, a document that has metadata but no document contents.

Document Attributes and Properties Document attributes, or metadata, describe document properties. Some attributes can be irrelevant to your application. The crawler plug-in creator must decide which document attributes should be extracted and saved. The plug-in also can be created such that the list of collected attributes are configurable. Oracle SES automatically registers attributes returned by the plug-in. The plug-in can decide which attributes to return for a document.

Library Path and Java Class Path Any other Java class needed by the plug-in should be included in the plug-in jar file. (You could add the paths for the additional jar files needed by the plug-in into the `Class-Path` of the `MANIFEST.MF` file within the plug-in jar file.) This is because Oracle SES automatically adds the plug-in jar file to the crawler Java class path, and Oracle SES does not let you add other class paths from the administration interface.

If the plug-in code also relies on a particular library file (for example, a .dll file on Windows or a .so file on UNIX), then the library path environment variable (for example, `PATH` on Windows or `LD_LIBRARY_PATH` on Linux) must contain the path to it. Make sure that Oracle is started from this environment. As the crawler is spawned by the Oracle process, it automatically inherits all environment variables from Oracle, including the library path.

Crawler Plug-in Restrictions The plug-in must handle mimetype rejection and large document rejection itself. For example, the plug-in should reject files it does not want to index based on its type or size, such as zip files. Also, plain text files, such as log files, can grow very large. Because the crawler reads HTML and plain text files into memory, it could run out of memory with very large files.

Crawler Plug-in Functionality

This section describes aspects of the crawler plug-in.

Source Registration Source registration is automated. After a source type is defined, any instance of that source type can be defined:

- Source name
- Description of the source; limit to 4000 bytes
- Source type ID
- Default language; default is 'en' (English)

- Parameter values; for example:

```
seed - http://www.oracle.com
depth - 8
```

Source Attribute Registration You can add new attributes to Oracle SES by providing the attribute name and the attribute data type. The data type can be string, number, or date. Attributes returned by an plug-in are automatically registered if they have not been defined.

User-Implemented Crawler Plug-in The crawler plug-in has the following requirements:

- The plug-in must be implemented in Java.
- The plug-in must support the Java plug-in APIs defined by Oracle SES.
- The plug-in must return the URL attributes and properties.
- The plug-in must decide which document attributes Oracle SES should keep. Any attribute not defined in Oracle SES is registered automatically.
- The plug-in can map attributes to source properties. For example, if an attribute "ID" is the unique ID of a document, then the plug-in should return (document_key, 4) where "ID" has been mapped to the property "document_key" and its value is 4 for this particular document.
- If the attribute LOV is available, then the plug-in returns them upon request.

Crawler Plug-in APIs and Classes The Crawler Plug-in API is a collection of classes and interfaces used to implement a crawler plug-in.

Table 6–3 Crawler Plug-in APIs and Classes

API/Class	Description
`CrawlerPluginManager`	This interface is used to generate the crawler plug-in instances. It provides general plug-in information for automatic plug-in registration on the administration page for defining user-defined source types. It has the control on which plug-in object (if more than one implementation is available) to return in `getCrawlerPlugin` call and how many instances of the plug-in to return. If only one instance is returned, then the plug-in implementation must handle multi-threading execution. The `CrawlingThreadService` object pass in is thread-specific as the invocation of each `getCrawlerPlugin` call is initiated by each thread.
`CrawlerPlugin`	This interface is used by the crawler plug-in to integrate with the Oracle SES crawler. The Oracle SES crawler loads the plug-in manager class and invokes the plug-in manager API to obtain the crawler plug-in instance. Each plug-in instance is run in the context of a thread execution.
`QueueService`	This API is implemented by the Oracle SES crawler and made available to the plug-in through the `GeneralService` object. This interface is used by the crawler plug-in to submit URL-related data to the crawler.

Table 6–3 (Cont.) Crawler Plug-in APIs and Classes

API/Class	Description
`DataSourceService`	This API is implemented by the Oracle SES crawler and made available to the plug-in through the `GeneralService` object.
	This interface is used by a crawler plug-in to manage the current crawled document set.
`GeneralService`	This API provides Oracle SES service and implemented interface objects to the plug-in. It is implemented by the Oracle SES crawler and made available through plug-in manager initialization.
	This interface is used by a crawler plug-in to obtain Oracle SES interface objects.
`CrawlingThreadService`	This interface is used by a crawler plug-in to perform crawler-related tasks. It has execution context specific to the crawling thread that invokes the plug-in `crawl()` method.
`DocumentMetadata`	This API holds a document's attributes and properties for processing and indexing.
	This interface is used by a crawler plug-in to submit URL-related data to the crawler.
`DocumentContainer`	This interface is used by a crawler plug-in to submit or retrieve document information.
`DocumentAcl`	This interface is used by a crawler plug-in to submit access control list (ACL) information for the document.
`ProcessingException`	This class encapsulates information about errors from processing plug-in requests.

URL Rewriter API

A URL rewriter is a user supplied Java module that implements the Oracle SES UrlRewriter Java interface. When activated, it is used by the crawler to filter and rewrite extracted URL links before they are inserted into the URL queue.

> **Note:** The URL Rewriter API is included as part of the Crawler Plug-in SDK. The URL Rewriter API is used for Web sources.

Web crawling generally consists of the following steps:

1. Get the next URL from the URL queue. (Web crawling stops when the queue is empty.)
2. Fetch the contents of the URL.
3. Extract URL links from the contents.
4. Insert the links into the URL queue.

The generated new URL link is subject to all existing boundary rules.

There are two possible operations that can be done on the extracted URL link:

- Filtering: removes the unwanted URL link
- Rewriting: transforms the URL link

URL Link Filtering

Users control what type of URL links are allowed to be inserted into the queue with the following mechanisms supported by the Oracle SES crawler:

- `robots.txt` file on the target Web site; for example, disallow URLs from the `/cgi` directory
- Hosts inclusion and exclusion rules; for example, only allow URLs from www.example.com
- File path inclusion and exclusion rules; for example, only allow URLs under the `/archive` directory
- Mimetype inclusion rules; for example, only allow HTML and PDF files
- Robots metatag `NOFOLLOW`; for example, do not extract any link from that page
- Black list URL; for example, URL explicitly singled out not to be crawled

With these mechanisms, only URL links that meet the filtering criteria are processed. However, there are other criteria that users might want to use to filter URL links. For example:

- Allow URLs with certain file name extensions
- Allow URLs only from a particular port number
- Disallow any PDF file if it is from a particular directory

The possible criteria could be very large, which is why it is delegated to a user-implemented module that can be used by the crawler when evaluating an extracted URL link.

URL Link Rewriting

For some applications, due to security reasons, the URL crawled is different from the one seen by the end user. For example, crawling is done on an internal Web site behind a firewall without security checking, but when queried by an end user, a corresponding mirror URL outside the firewall must be used.

A *display URL* is a URL string used for search result display. This is the URL used when users click the search result link. An *access URL* is a URL string used by the crawler for crawling and indexing. An access URL is optional. If it does not exist, then the crawler uses the display URL for crawling and indexing. If it does exist, then it is used by the crawler instead of the display URL for crawling.

For regular Web crawling, there are only display URLs available. But in some situations, the crawler needs an access URL for crawling the internal site while keeping a display URL for the external use. For every internal URL, there is an external mirrored one.

For example:

```
http://www.example-qa.us.com:9393/index.html
http://www.example.com/index.html
```

When the URL link `http://www.example-qa.us.com:9393/index.html` is extracted and before it is inserted into the queue, the crawler generates a new display URL and a new access URL for it:

Access URL:

```
http://www.example-qa.us.com:9393/index.html
```

Display URL:

```
http://www.example.com/index.html
```

The extracted URL link is rewritten, and the crawler crawls the internal Web site without exposing it to the end user.

Another example is when the links that the crawler picks up are generated dynamically and can be different (depending on referencing page or other factor) even though they all point to the same page. For example:

```
http://compete3.example.com/rt/rt.wwv_media.show?p_type=text&p_id=4424&p_
currcornerid=281&p_textid=4423&p_language=us

http://compete3.example.com/rt/rt.wwv_media.show?p_type=text&p_id=4424&p_
currcornerid=498&p_textid=4423&p_language=us
```

Because the crawler detects different URLs with the same contents only when there is sufficient number of duplication, the URL queue could grow to a huge number of URLs, causing excessive URL link generation. In this situation, allow "normalization" of the extracted links so that URLs pointing to the same page have the same URL. The algorithm for rewriting these URLs is application dependent and cannot be handled by the crawler in a generic way.

When a URL link goes through a rewriter, there are the following possible outcomes:

- The link is inserted with no changes made to it.
- The link is discarded; it is not inserted.
- A new display URL is returned, replacing the URL link for insertion.
- A display URL and an access URL are returned. The display URL might or might not be identical to the URL link.

Creating and Using a URL Rewriter

Follow these steps to create and use a URL rewriter:

1. Create a new Java file implementing the UrlRewriter interface `open`, `close`, and `rewrite` methods. A rewriter, `SampleRewriter.java`, is available for reference under `$ORACLE_HOME/search/sample/`.
2. Compile the rewriter Java file into a class file. For example:

   ```
   $ORACLE_HOME/jdk/bin/javac -classpath $ORACLE_HOME/search/lib/search.jar
   SampleRewriter.java
   ```

3. Package the rewriter class file into a jar file under the `$ORACLE_HOME/search/lib/agent/` directory. For example:

   ```
   $ORACLE_HOME/jdk/bin/jar cv0f $ORACLE_HOME/search/lib/agent/sample.jar
   SampleRewriter.class
   ```

4. Enable the UrlRewriter option and specify the rewriter class name and jar file name (for example, `SampleRewriter` and `sample.jar`) in the administration tool **Home - Sources - Crawling Parameters** page of an existing Web source
5. Crawl the target Web source by launching the corresponding schedule. The crawler log file confirms the use of the URL rewriter with the message *Loading URL rewriter "SampleRewriter"...*

Note: URL rewriting is available for Web sources only.

See Also:

- *Oracle Secure Enterprise Search Java API Reference* for the API (`oracle.search.crawler` package)
- The URL rewriter `SampleRewriter.java` under `$ORACLE_HOME/search/sample/`

Query-time Authorization API

Query-time authorization allows an Oracle Secure Enterprise Search (SES) administrator to associate a Java class with a source that will, at search time, validate every document fetched out of the Oracle SES repository belonging to the protected source. The filter class can dynamically check access rights to make sure that the current search user has the credentials to view each document.

This authorization model can be applied to any source other than self service or federated sources. Besides acting as the sole provider of access control for a source, it can also be used as a post-filter. For example, a source can be stamped with a more generic ACL, while query-time authorization can be used to fine tune the results.

Overview of Query-time Authorization

Query-time authorization has the following characteristics:

- It allows dynamic access control at search time compared to more static ACL stamping.
- It filters documents returned to a search user.
- It controls the Browse functionality to determine whether a folder is visible to a search user
- Optionally, it allows pruning of an entire source from the results to reduce performance costs of filtering each document individually
- It is applicable to all source types except self service and federated sources

Query-time filtering is handled by class implementations of the `QueryTimeFilter` interface.

Filtering Document Access

Filtering document access is handled by the `filterDocuments` method of the `QueryTimeFilter` interface. The most common situation for filtering will occur with a search request, in which this method will be invoked with batches of documents from the result list. Based on the values returned by this method, all, some, or none of the documents might be removed from the results returned to the search user.

Access of individual documents is also controlled. For example, viewing a cached copy of a document or accessing the in-links and out-links will require a call into `filterDocuments` to determine the authorization for the search user.

Filtering Folder Browsing

The `QueryTimeFilter` implementation is also responsible for controlling the access to, and visibility of folders in, the Browse application. If a folder belongs to a source protected by a query-time filter, then the folder name in the **Browse** page will not have a document count listed next to it. Instead, the folder will show a **view_all** link.

For performance reasons, it could be costly to determine the exact number of documents visible to the current search user for every query-time filtered folder

displayed on a Browse page. This task would require that every document in every folder be processed by the filter in order to calculate the total number of documents available for each folder. To prevent this comprehensive and potentially time-consuming operation, document counts are not used. Instead, folder visibility is explicitly determined by the query-time filter.

Based on the results from the `filterBrowseFolders` method, a folder might be hidden or shown in the Browse page. This result also controls access to the single folder browsing page, which displays the documents contained in a folder.

If the security of folder names is not a concern for a particular source, then the `filterBrowseFolders` method can blindly authorize all folders to be visible in the Browse application. After a folder is selected, the document list is still filtered through the `filterDocuments` method. This strategy should not be employed if folder names could reveal sensitive information.

If security is very critical, then it might be easiest to hide all folders for browsing. The documents from the source will still be available for search queries from the Basic and Advanced Search boxes, but a user will not be able to browse the source in the **Browse** pages of the search application.

Limitations of folder filtering:

- The `filterBrowseFolders` method does not implicitly restrict access to subfolders. For example, if folder `/Miscellaneous/www.example.com/private` is hidden for a search user, then it is still possible for that user to view any subfolder, such as `/Miscellaneous/www.example.com/private/a/b`, if that subfolder is not also explicitly filtered out by this method. It would be possible to view this subfolder if the user followed a bookmark or outside link directly to the authorized subfolder in the Browse application.
- This method does not affect functionality outside of the Browse application. This is not a generic folder pruning method. Search queries and document retrieval outside of the Browse application are only affected by the `filterDocuments` and `pruneSource` methods.

Pruning Access to an Entire Source

The `QueryTimeFilter` interface provides the ability to determine access privileges at the source level. This is achieved through calls to the `pruneSource` method. This method can be called in situations where there are a large number of documents or folders to be filtered. Authorizing or unauthorizing the entire source for a given user could provide a large performance gain over filtering each document individually.

The implementation of `QueryTimeFilter` must not rely on this method to secure access to documents or folders. This method is strictly an optimization feature. There is no guarantee that this will ever be invoked for any particular search request or document access. For example, when performing authorization for a single document, Oracle SES may call the `filterDocuments` method directly without invoking this method at all. Therefore, the `filterDocuments` and `filterBrowseFolders` methods must be implemented to provide full security in the absence of pruning.

Determining the Authenticated User

A query-time filter is free to define a search user's access privileges to sources and documents based on any criteria available. For example, a filter could be written to deny access to a source depending on the time of day.

In most cases, however, a filter will impose restrictions based on the authenticated user for that search request. The Oracle SES authenticated user name for a request is contained in the `RequestInfo` object. The steps for accessing this user name value depend on whether the request originated from the JSP search application or the Oracle SES Web Services interface. For either type of request, the key used to access the authenticated user name is the string value `AUTH_USER`.

This sample implementation of the `QueryTimeFilter.getCurrentUserName` method illustrates how to retrieve the current authenticated user from either a JSP or Web Services request:

```
public String getCurrentUserName( RequestInfo req )
  throws QueryTimeFilterException
{
  HttpServletRequest servReq = req.getHttpRequest();
  Map sessCtx = req.getSessionContext();
  String user = null;

  if( servReq != null )
  {
    // JSP request
    HttpSession session = servReq.getSession();
    if( session != null )
      user = ( String ) session.getAttribute( "AUTH_USER" );
  }
  else if( sessCtx != null )
  {
    // Web Service request
    user = ( String ) sessCtx.get( "AUTH_USER" );
  }

  return user;
}
```

See Also: "Authentication Methods" on page 4-6

Query-time Authorization Interfaces and Exceptions

The `oracle.search.query.qta` package contains all interfaces and exceptions in the Query-time Authorization API.

To write a query-time authorization filter, implement the `QueryTimeFilter` interface. The methods in this interface may throw instances of the `QueryTimeFilterException` exception.

Objects that implement the `RequestInfo`, `DocumentInfo`, and `FolderInfo` interfaces are passed in as arguments for filtering, but these interfaces do not need to be implemented by the filter writer.

The API contains the following interfaces and exceptions:

Table 6–4 Query-time Authorization Interfaces and Exceptions

Interface/Exception	Description
`QueryTimeFilter`	This interface filters search results and access to document information at search time. If an object implementing this interface has been assigned to a source, then any search results or other retrieval of documents belonging to the source are passed through this filter before being presented to the end user.
`QueryTimeFilterException`	This exception is thrown by methods in the `QueryTimeFilter` interface to indicate that a failure has occurred.
`RequestInfo`	This interface represents information about a request that can be passed to a `QueryTimeFilter` for filtering out documents, folders, or entire sources.
`DocumentInfo`	This interface represents information about a document that can be passed to a `QueryTimeFilter` for filtering out documents.
`FolderInfo`	This interface represents information about a folder that can be passed to a `QueryTimeFilter` to control folder browsing.

See Also: *Oracle Secure Enterprise Search Java API Reference* for the `oracle.search.query.qta` package

Thread-safety of the Filter Implementation

Classes that implement the `QueryTimeFilter` interface should be designed to persist for the lifetime of a running Oracle SES search application. A single instance of `QueryTimeFilter` will generally handle multiple concurrent requests from different search end users. Therefore, the `filterDocuments`, `pruneSource`, `filterBrowseFolders`, and `getCurrentUserName` methods in this class must be both reentrant and thread-safe.

Compiling and Packaging the Query-time Filter

To compile your query-time filter class, you will need to include at least the two following files in the Java CLASSPATH. These files can be found in the Oracle SES server directory.

- `$ORACLE_HOME/search/lib/search_query.jar`
- `$ORACLE_HOME/jlib/servlet.jar`

It is recommended to build a jar file containing your `QueryTimeFilter` class (or classes) and any supporting Java classes. This jar file should be placed in a secure location for access by the Oracle SES server. If this jar file is compromised, then the security of document access in the search server can be compromised.

Your query-time filter might require other class or jar files that are not included in the jar file you build and are not located in the Oracle SES class path. If so, these files should be added to the Class-Path attribute of the JAR file manifest. This manifest file should be included in the jar file you build.

If Oracle SES cannot locate a class used by a `QueryTimeFilter` during run-time, then an error message will be written to the log file and all documents from that source will be filtered out for the search request being processed.

See Also:

`http://java.sun.com/j2se/1.4.2/docs/guide/jar/jar.html` for more information about JAR file manifests.

Sample Query-time Filter Files

The sample query-time filter files are posted on the Oracle SES home page:

`http://www.oracle.com/technology/products/oses/index.html`

You can view the filter source code using a text editor.

See Also: *Oracle Secure Enterprise Search Java API Reference*

A

URL Crawler Status Codes

The crawler uses a set of codes to indicate the result of the crawled URL. Besides the standard HTTP status code, it uses its own code for non-HTTP related situations.

Only URLs with status 200 will be indexed. If the record exists in `EQ$URL` but the status is something other than 200, then the crawler encountered an error trying to fetch the document. A status of less than 600 maps directly to the HTTP status code.

The following table lists the URL status codes, document container codes used by the crawler plug-in, and EQG codes.

Code	Description	Document Container Code	EQG Codes
200	URL OK	`STATUS_OK_FOR_INDEX`	N/A
400	Bad request	`STATUS_BAD_REQUEST`	30009
401	Authorization required	`STATUS_AUTH_REQUIRED`	30007
402	Payment required		30011
403	Access forbidden	`STATUS_ACCESS_FORBIDDEN`	30010
404	Not found	`STATUS_NOTFOUND`	30008
405	Method not allowed		30012
406	Not acceptable		30013
407	Proxy authentication required	`STATUS_PROXY_REQUIRED`	30014
408	Request timeout	`STATUS_REQUEST_TIMEOUT`	30015
409	Conflict		30016
410	Gone		30017
414	Request URI too large		30066
500	Internal server error	`STATUS_SERVER_ERROR`	10018
501	Not implemented		10019
502	Bad gateway	`STATUS_BAD_GATEWAY`	10020
503	Service unavailable	`STATUS_FETCH_ERROR`	10021
504	Gateway timeout		10022
505	HTTP version not supported		10023
902	Timeout reading document	`STATUS_READ_TIMEOUT`	30057
903	Filtering failed	`STATUS_FILTER_ERROR`	30065

Code	Description	Document Container Code	EQG Codes
904	Out of memory error	STATUS_OUT_OF_MEMORY	30003
905	IOEXCEPTION in processing URL	STATUS_IO_EXCEPTION	30002
906	Connection refused	STATUS_CONNECTION_ REFUSED	30025
907	Socket bind exception		30079
908	Filter not available		30081
909	Duplicate document detected		30082
910	Duplicate document ignored	STATUS_DUPLICATE_DOC	30083
911	Empty document	STATUS_EMPTY_DOC	30106
951	URL not indexed (this can happen if robots.txt specifies that a certain document should not be indexed)	STATUS_OK_BUT_NO_INDEX	N/A
952	URL crawled	STATUS_OK_CRAWLED	N/A
953	Metatag redirection		N/A
954	HTTP redirection		30000
955	Black list URL		N/A
956	URL is not unique		31017
957	Sentry URL (URL as a place holder)		N/A
958	Document read error	STATUS_CANNOT_READ	30173
959	Form login failed	STATUS_LOGIN_FAILED	30183
960	Document size too big, ignored	STATUS_DOC_SIZE_TOO_BIG	30209
1001	Datatype is not TEXT/HTML		30001
1002	Broken network data stream		30004
1003	HTTP redirect location does not exist		30005
1004	Bad relative URL		30006
1005	HTTP error		30024
1006	Error parsing HTTP header		30058
1007	Invalid URL table column name		30067
1009	Binary document reported as text document		30126
1010	Invalid display URL		30112
1011	Invalid XML from OracleAS Portal	PORTAL_XMLURL_FAIL	31011
1020-1024	URL is not reachable. The status starts at 1020, and it increases by one with each try. After five tries (if it reaches 1025), the URL is deleted.		N/A

Code	Description	Document Container Code	EQG Codes
1111	URL remained in the queue even after a successful crawl. This indicates that the crawler had a problem processing this document. You could investigate the URL by crawling it in a separate source to check for errors in the crawler log.		N/A

B

Error Messages

The crawler uses a set of messages to log the crawling activities.

The following table lists the most common crawler error messages.

Message ID	Message	Comment	Action
30025	{0}: Connection refused	The Web site refuses the URL access request.	Check the network setup environment of the machine running the crawler.
30027	Not allowed URL: {0}	A URL link violates boundary rules and is discarded.	Confirm that the URL indeed can be ignored.
30030	Malformed URL: {0}	The URL is not properly formed.	Verify the URL.
30031	Excluded by ROBOTS.TXT: {0}	The robots.txt rule from the Web site of the URL does not allow the URL to be crawled.	Configure the crawler to ignore robots rule only when you are managing the target Web site. This is done on the **Home - Sources - Crawling Parameters** page.
30040	Ignore URL: {0}	Redirection to this URL is not allowed by boundary rule.	Confirm that the URL indeed should be ignored.
30041	{0}: excluded by MIME type inclusion rule, URL is {1}	The content type of the URL is not in MIME type inclusion list.	Check if the specified content type should be included.
30054	Excessively long URL: {0}	The URL string is too long, and the URL is ignored.	N/A
30057	{0}: timeout reading document	The target Web site is too slow sending page content.	Increase the crawler timeout threshold from the crawler configuration page. The default is 30 seconds.
30083	{0}: Duplicate document ignored	A document with the same content has been seen before within the same crawl session. This could be an indication of URL looping; that is, a generation of different URLs pointing back to the same page.	Check if the URL is generated correctly. If necessary, disable indexing dynamic URLs. This is done on the **Home - Sources - Crawling Parameters** page.

Message ID	Message	Comment	Action
30126	Binary document reported as text document: "{0}"	A binary file has been sent by the Web site as a text document. In most cases, the URL in question is not a binary format text document, like pdf.	Correct the Web site content type setting for the URL, if possible.
30188	Login form not specified for "{0}"	Unable to perform HTML form login, because the name of the form is not set. In general, the name of the form should be automatically set by the crawler.	Identify the URL of the login page, and check whether this is a regular HTML form login page or a SSO login page. Report the problem to Oracle support.
30199	Encountered an error while responding to the following HTTP authentication request: [{0}]	Unable to authenticate through the target URL.	Verify if the authentication request is basic authentication or digest authentication. Also confirm the provided authentication credentials.
30201	Missing authentication credentials	Authentication data is not available to access the URL.	Check the type of authentication needed and provide it through the source customization page
30206	Ignoring "{0}" due to host (or redirected host) connection problem	The crawler is unable to contact the server of the URL.	Verify that the Web site in question is up and try to recrawl.
30209	Document size ({0}) too big, ignored: {1}	Document size exceeds the default limit of 10 megabytes.	Increase the document size limit on the **Global Settings - Crawler Configuration** page.
30215	Excluded by crawling depth limit({0}): {1}	Previously crawled URL is excluded due to newly reduced crawling depth limit.	Confirm that the depth limit is correct.
30782	Invalid document attribute {0} - ignored	Some of the attribute picked up from the document is not defined for the source. It is ignored.	Most likely this is safe to ignore, unless you know that this particular attribute should be defined for this source. In that case, contact Oracle Support.

C

WSDL Specification

The Web Services Description Language (WSDL) is an XML format for describing network services containing RPC-oriented and message-oriented information. Programmers or automated development tools can create WSDL files to describe a service and can make the description available over the Internet. Client-side programmers and development tools can use published WSDL specifications to obtain information about available Web Services and to build and create proxies or program templates that access available services.

This appendix provides the WSDL description of the Oracle SES Web Services API.

See Also: "Oracle Secure Enterprise Search Web Services API" on page 6-1

```
<definitions name="OracleSearchService"

targetNamespace="http://oracle.search.query.webservice/OracleSearchService.wsdl"
             xmlns:typens="http://oes.oracle.com/OracleSearch"

xmlns:tns="http://oracle.search.query.webservice/OracleSearchService.wsdl"
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
             xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
             xmlns="http://schemas.xmlsoap.org/wsdl/">

  <!-- Types for search - result elements, directory categories -->

  <types>
    <xsd:schema
            xmlns="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://oes.oracle.com/OracleSearch"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
>

      <xsd:complexType name="OracleSearchResult">
        <xsd:all>
  <xsd:element name="returnCount"       type="xsd:boolean"/>
          <xsd:element name="estimatedHitCount" type="xsd:int"/>
          <xsd:element name="dupRemoved"        type="xsd:boolean"/>
          <xsd:element name="dupMarked"         type="xsd:boolean"/>
          <xsd:element name="resultElements"    type="typens:ResultElementArray"/>
          <xsd:element name="suggestedLinks"    type="typens:SuggestedLinkArray"/>
          <xsd:element name="query"             type="xsd:string"/>
          <xsd:element name="altKeywords"       type="xsd:string"/>
          <xsd:element name="startIndex"        type="xsd:int"/>
```

```
            <xsd:element name="docsReturned"      type="xsd:int"/>

          </xsd:all>
        </xsd:complexType>

        <xsd:complexType name="ResultElement">
          <xsd:all>
            <xsd:element name="author" type="xsd:string"/>
            <xsd:element name="description" type="xsd:string"/>
            <xsd:element name="url" type="xsd:string"/>
            <xsd:element name="snippet" type="xsd:string"/>
            <xsd:element name="title" type="xsd:string"/>
            <xsd:element name="lastModified" type="xsd:date"/>
            <xsd:element name="mimetype" type="xsd:string"/>
            <xsd:element name="score" type="xsd:int"/>
            <xsd:element name="docID" type="xsd:int"/>
            <xsd:element name="language" type="xsd:string"/>
            <xsd:element name="contentLength" type="xsd:int"/>
            <xsd:element name="signature" type="xsd:long"/>
            <xsd:element name="infoSourceID" type="xsd:string"/>
            <xsd:element name="infoSourcePath" type="xsd:string"/>
            <xsd:element name="groups" type="typens:DataGroupArray"/>
            <xsd:element name="isDuplicate" type="xsd:boolean"/>
            <xsd:element name="hasDuplicate" type="xsd:boolean"/>
            <xsd:element name="fedID" type="xsd:string"/>
            <xsd:element name="customAttributes"
type="typens:CustomAttributeArray"/>
          </xsd:all>
        </xsd:complexType>

        <xsd:complexType name="ResultElementArray">
          <xsd:complexContent>
            <xsd:restriction base="soapenc:Array">
               <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:ResultElement[]"/>
            </xsd:restriction>
          </xsd:complexContent>
        </xsd:complexType>

        <xsd:complexType name="CustomAttribute">
          <xsd:all>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="value" type="xsd:string"/>
          </xsd:all>
        </xsd:complexType>

        <xsd:complexType name="CustomAttributeArray">
          <xsd:complexContent>
            <xsd:restriction base="soapenc:Array">
               <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:CustomAttribute[]"/>
            </xsd:restriction>
          </xsd:complexContent>
        </xsd:complexType>

        <xsd:complexType name="SuggestedLink">
          <xsd:all>
            <xsd:element name="title"      type="xsd:string"/>
            <xsd:element name="url"     type="xsd:string"/>
          </xsd:all>
```

```
      </xsd:complexType>

      <xsd:complexType name="SuggestedLinkArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
             <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:SuggestedLink[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="DataGroup">
        <xsd:all>
          <xsd:element name="groupID" type="xsd:int"/>
          <xsd:element name="groupName" type="xsd:string"/>
          <xsd:element name="groupDisplayName" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="DataGroupArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="item"
type="typens:DataGroup"/>
    </xsd:sequence>
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:DataGroup[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="Language">
        <xsd:all>
          <xsd:element name="languageName" type="xsd:string"/>
          <xsd:element name="languageDisplayName" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="LanguageArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="item"
type="typens:Language"/>
    </xsd:sequence>
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:Language[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="SessionContextElement">
        <xsd:all>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="value" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="SessionContextElementArray">
```

```
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="item"
type="typens:SessionContextElement"/>
    </xsd:sequence>
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:SessionContextElement[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>


      <xsd:complexType name="FilterArray">
         <xsd:complexContent>
   <xsd:restriction base="soapenc:Array">
     <xsd:sequence>
       <xsd:element maxOccurs="unbounded" minOccurs="0" name="item"
type="typens:Filter"/>
     </xsd:sequence>
     <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:Filter[]"/>
   </xsd:restriction>
         </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="Filter">
   <xsd:all>
          <xsd:element name="attributeId" type="xsd:int"/>
          <xsd:element name="attributeType"   type="xsd:string"/>
          <xsd:element name="operator" type="xsd:string"/>
          <xsd:element name="attributeValue"type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>


      <xsd:complexType name="StringArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="IntArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:int[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>


      <xsd:complexType name="Status">
        <xsd:all>
          <xsd:element name="status"   type="xsd:string"/>
          <xsd:element name="message"  type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="Node">
```

```
        <xsd:all>
          <xsd:element name="id"            type="xsd:string"/>
          <xsd:element name="fedId"         type="xsd:string"/>
          <xsd:element name="name"          type="xsd:string"/>
          <xsd:element name="docCount"      type="xsd:int"/>
          <xsd:element name="hasChildren"   type="xsd:boolean"/>
          <xsd:element name="fullpath"      type="typens:StringArray"/>
          <xsd:element name="fullpathIds"   type="typens:StringArray"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="NodeArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:Node[]"/>
  </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="Attribute">
        <xsd:all>
    <xsd:element name="id" type="xsd:int"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="displayName" type="xsd:string"/>
    <xsd:element name="type" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="AttributeArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:Attribute[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="AttributeLOVElement">
        <xsd:all>
          <xsd:element name="value" type="xsd:string"/>
          <xsd:element name="displayValue" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="AttributeLOVElementArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:AttributeLOVElement[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>
  </types>


  <!-- Messages for Oracle Enterprise Search Web Services APIs -->
```

```
<message name="doOracleSearch">
  <part name="query"              type="xsd:string"/>
  <part name="startIndex"         type="xsd:int"/>
  <part name="docsRequested"      type="xsd:int"/>
  <part name="dupRemoved"         type="xsd:boolean"/>
  <part name="dupMarked"          type="xsd:boolean"/>
  <part name="groups"             type="typens:DataGroupArray"/>
  <part name="queryLang"          type="xsd:string"/>
  <part name="docLang"            type="xsd:string"/>
  <part name="returnCount"        type="xsd:boolean"/>
  <part name="filterConnector"    type="xsd:string"/>
  <part name="filters"     type="typens:FilterArray"/>
  <part name="fetchAttributes"    type="typens:IntArray"/>
</message>

<message name="doOracleSearchResponse">
  <part name="return"         type="typens:OracleSearchResult"/>
</message>

<message name="doOracleSimpleSearch">
  <part name="query"              type="xsd:string"/>
  <part name="startIndex"         type="xsd:int"/>
  <part name="docsRequested"      type="xsd:int"/>
  <part name="dupRemoved"         type="xsd:boolean"/>
  <part name="dupMarked"          type="xsd:boolean"/>
  <part name="returnCount"        type="xsd:boolean"/>
</message>

<message name="doOracleSimpleSearchResponse">
  <part name="return"         type="typens:OracleSearchResult"/>
</message>

<message name="doOracleBrowseSearch">
  <part name="query"              type="xsd:string"/>
  <part name="nodeID"             type="xsd:string"/>
  <part name="fedID"              type="xsd:string"/>
  <part name="startIndex"         type="xsd:int"/>
  <part name="docsRequested"      type="xsd:int"/>
  <part name="dupRemoved"         type="xsd:boolean"/>
  <part name="dupMarked"          type="xsd:boolean"/>
  <part name="queryLang"          type="xsd:string"/>
  <part name="docLang"            type="xsd:string"/>
  <part name="returnCount"        type="xsd:boolean"/>
  <part name="fetchAttributes"    type="typens:IntArray"/>
</message>

<message name="doOracleBrowseSearchResponse">
  <part name="return"         type="typens:OracleSearchResult"/>
</message>

<message name="proxyLoginRequest">
  <part name="username"            type="xsd:string"/>
  <part name="password"            type="xsd:string"/>
  <part name="searchUser"            type="xsd:string"/>
</message>

<message name="loginRequest">
  <part name="username"            type="xsd:string"/>
  <part name="password"            type="xsd:string"/>
</message>
```

```
<message name="loginResponse">
  <part name="return"type="typens:Status"/>
</message>

<message name="logoutRequest">
</message>

<message name="logoutResponse">
  <part name="return"type="typens:Status"/>
</message>

<message name="setSessionContextRequest">
  <part name="sessionContext"     type="typens:SessionContextElementArray"/>
</message>

<message name="setSearchUserRequest">
  <part name="username"            type="xsd:string"/>
</message>

<message name="setSessionContextResponse">
  <part name="return"             type="typens:Status"/>
</message>

<message name="setSearchUserResponse">
  <part name="return"type="typens:Status"/>
</message>

<message name="getCachedPage">
  <part name="query"            type="xsd:string"/>
  <part name="docID"            type="xsd:int"/>
  <part name="fedID"            type="xsd:string"/>
</message>

<message name="getCachedPageResponse">
  <part name="return"           type="xsd:base64Binary"/>
</message>


<message name="getInLinksRequest">
  <part name="docID"          type="xsd:int"/>
  <part name="maxNum"          type="xsd:int"/>
  <part name="fedID"            type="xsd:string"/>
</message>

<message name="getInLinksResponse">
  <part name="return"           type="typens:StringArray"/>
</message>

<message name="getOutLinksRequest">
  <part name="docID"            type="xsd:int"/>
  <part name="maxNum"           type="xsd:int"/>
  <part name="fedID"            type="xsd:string"/>
</message>

<message name="getOutLinksResponse">
  <part name="return"           type="typens:StringArray"/>
</message>

<message name="submitUrlRequest">
```

```
   <part name="Url"             type="xsd:string"/>
</message>

<message name="submitUrlResponse">
   <part name="return"          type="typens:Status"/>
</message>

<message name="getInfoSourceNodesRequest">
  <part name="parentNodeID"type="xsd:string"/>
  <part name="fedID"        type="xsd:string"/>
  <part name="locale"        type="xsd:string"/>
</message>

<message name="getInfoSourceNodesResponse">
  <part name="nodes"    type="typens:NodeArray"/>
</message>

<message name="getInfoSourceAncestorNodesRequest">
  <part name="nodeID"type="xsd:string"/>
  <part name="locale"type="xsd:string"/>
</message>

<message name="getInfoSourceAncestorNodesResponse">
  <part name="nodes"    type="typens:NodeArray"/>
</message>

<message name="getInfoSourceNodeRequest">
  <part name="nodeID"type="xsd:string"/>
  <part name="fedID"type="xsd:string"/>
  <part name="locale"type="xsd:string"/>
</message>

<message name="getInfoSourceNodeResponse">
  <part name="node"    type="typens:Node"/>
</message>


<message name="getLanguagesRequest">
  <part name="locale"        type="xsd:string"/>
</message>

<message name="getLanguagesResponse">
  <part name="return"         type="typens:LanguageArray"/>
</message>

<message name="getDataGroupsRequest">
  <part name="locale"        type="xsd:string"/>
</message>

<message name="getDataGroupsResponse">
  <part name="groups"         type="typens:DataGroupArray"/>
</message>

<message name="getAttributesRequest">
  <part name="locale"        type="xsd:string"/>
  <part name="groups"        type="typens:DataGroupArray"/>
  <part name="groupConnector"         type="xsd:string"/>
</message>

<message name="getAttributesResponse">
```

```
  <part name="return"         type="typens:AttributeArray"/>
</message>

<message name="getAllAttributesRequest">
  <part name="locale"        type="xsd:string"/>
</message>

<message name="getAllAttributesResponse">
  <part name="return"         type="typens:AttributeArray"/>
</message>

<message name="getAttributeLOVRequest">
  <part name="attribute"type="typens:Attribute"/>
  <part name="locale"        type="xsd:string"/>
</message>

<message name="getAttributeLOVResponse">
  <part name="return"         type="typens:AttributeLOVElementArray"/>
</message>

<message name="logUserClickRequest">
  <part name="queryID"type="xsd:int"/>
  <part name="urlID"        type="xsd:int"/>
  <part name="infoSourceID"        type="xsd:int"/>
  <part name="position"        type="xsd:int"/>
  <part name="fedID"                type="xsd:string"/>
</message>

<message name="logUserClickResponse">
  <part name="url"         type="xsd:string"/>
</message>

<!-- Port for Oracle Enterprise Search Web Services APIs, "OracleSearch" -->

<portType name="OracleSearchPort">

  <operation name="proxyLogin">
    <input message="tns:proxyLoginRequest"/>
    <output message="tns:loginResponse"/>
  </operation>

  <operation name="login">
    <input message="tns:loginRequest"/>
    <output message="tns:loginResponse"/>
  </operation>

  <operation name="logout">
    <input message="tns:logoutRequest"/>
    <output message="tns:logoutResponse"/>
  </operation>

  <operation name="setSessionContext">
    <input message="tns:setSessionContextRequest"/>
    <output message="tns:setSessionContextResponse"/>
  </operation>

  <operation name="setSearchUser">
    <input message="tns:setSearchUserRequest"/>
    <output message="tns:setSearchUserResponse"/>
  </operation>
```

```
    <operation name="getCachedPage">
      <input message="tns:getCachedPage"/>
      <output message="tns:getCachedPageResponse"/>
    </operation>

    <operation name="doOracleSearch">
      <input message="tns:doOracleSearch"/>
      <output message="tns:doOracleSearchResponse"/>
    </operation>

    <operation name="doOracleSimpleSearch">
      <input message="tns:doOracleSimpleSearch"/>
      <output message="tns:doOracleSimpleSearchResponse"/>
    </operation>

    <operation name="doOracleBrowseSearch">
      <input message="tns:doOracleBrowseSearch"/>
      <output message="tns:doOracleBrowseSearchResponse"/>
    </operation>

    <operation name="getDataGroups">
      <input message="tns:getDataGroupsRequest"/>
      <output message="tns:getDataGroupsResponse"/>
    </operation>

    <operation name="getAttributes">
<input message="tns:getAttributesRequest"/>
 <output message="tns:getAttributesResponse"/>
    </operation>

    <operation name="getAllAttributes">
<input message="tns:getAllAttributesRequest"/>
 <output message="tns:getAllAttributesResponse"/>
    </operation>

    <operation name="getAttributeLOV">
<input message="tns:getAttributeLOVRequest"/>
<output message="tns:getAttributeLOVResponse"/>
    </operation>

    <operation name="getLanguages">
<input message="tns:getLanguagesRequest"/>
<output message="tns:getLanguagesResponse"/>
    </operation>

    <operation name="getInLinks">
<input message="tns:getInLinksRequest"/>
<output message="tns:getInLinksResponse"/>
    </operation>

    <operation name="getOutLinks">
<input message="tns:getOutLinksRequest"/>
<output message="tns:getOutLinksResponse"/>
    </operation>

    <operation name="submitUrl">
<input message="tns:submitUrlRequest"/>
<output message="tns:submitUrlResponse"/>
    </operation>
```

```
    <operation name="getInfoSourceNodes">
<input message="tns:getInfoSourceNodesRequest"/>
<output message="tns:getInfoSourceNodesResponse"/>
    </operation>

    <operation name="getInfoSourceAncestorNodes">
    <input message="tns:getInfoSourceAncestorNodesRequest"/>
<output message="tns:getInfoSourceAncestorNodesResponse"/>
    </operation>

    <operation name="getInfoSourceNode">
<input message="tns:getInfoSourceNodeRequest"/>
<output message="tns:getInfoSourceNodeResponse"/>
    </operation>

    <operation name="logUserClick">
<input message="tns:logUserClickRequest"/>
<output message="tns:logUserClickResponse"/>
    </operation>


  </portType>

  <!-- Binding for Oracle Enterprise Search Web Services APIs - RPC, SOAP over
HTTP -->

  <binding name="OracleSearchBinding" type="tns:OracleSearchPort">
    <soap:binding style="rpc"
                  transport="http://schemas.xmlsoap.org/soap/http"/>

    <operation name="setSearchUser">
      <soap:operation soapAction="http://oes.oracle.com/OracleSearch/action"/>
      <input>
        <soap:body use="encoded"
                   namespace="http://oes.oracle.com/OracleSearch"
                   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
                   namespace="http://oes.oracle.com/OracleSearch"
                   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="proxyLogin">
      <soap:operation soapAction="http://oes.oracle.com/OracleSearch/action"/>
      <input>
        <soap:body use="encoded"
                   namespace="http://oes.oracle.com/OracleSearch"
                   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
                   namespace="http://oes.oracle.com/OracleSearch"
                   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="login">
```

```
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </input>
  <output>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </output>
</operation>

<operation name="logout">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </input>
  <output>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </output>
</operation>

<operation name="setSessionContext">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </input>
  <output>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </output>
</operation>

<operation name="getCachedPage">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </input>
  <output>
    <soap:body use="encoded"
               namespace="OracleSearchService"
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </output>
</operation>

<operation name="doOracleSearch">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="encoded"
               namespace="OracleSearchService"
```

```
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
                    namespace="OracleSearchService"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="doOracleSimpleSearch">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="encoded"
                    namespace="OracleSearchService"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
                    namespace="OracleSearchService"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="doOracleBrowseSearch">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="encoded"
                    namespace="OracleSearchService"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
                    namespace="OracleSearchService"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="getDataGroups">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
              namespace="OracleSearchService"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>

    <operation name="getAttributes">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
```

```
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>

    <operation name="getAllAttributes">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>

    <operation name="getAttributeLOV">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>

    <operation name="getLanguages">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="getInLinks">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
       encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="getOutLinks">
```

```
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
       encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="submitUrl">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="getInfoSourceNodes">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="getInfoSourceAncestorNodes">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>
    <operation name="getInfoSourceNode">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
```

```
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>

    <operation name="logUserClick">
<soap:operation soapAction=""/>
<input>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</input>
<output>
  <soap:body use="encoded"
 namespace="OracleSearchService"
 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
</output>
    </operation>


  </binding>

  <!-- Endpoint for Oracle Enterprise Search Web Services APIs -->
  <service name="OracleSearchService">
    <port name="OracleSearchPort" binding="tns:OracleSearchBinding">
      <soap:address location="http://myserver:7777/search/query/OracleSearch"/>
    </port>
  </service>

</definitions>
```

D

LDIF Files

This appendix lists the LDIF files necessary to set up secure search with Oracle Calendar and Oracle Content Database sources.

See Also: "Setting Up Secure Oracle Calendar Sources" on page 5-6 and "Setting Up Secure Oracle Content Database Sources" on page 5-8

calPlugin.ldif

```
# create product
dn: cn=oses,cn=products,cn=oraclecontext
changetype: add
objectClass: orclContainer
objectClass: top

# create application entity
dn: orclapplicationcommonname=ocscalplugin,cn=oses,cn=products,cn=oraclecontext
changetype: add
objectClass: orclApplicationEntity
objectClass: top
orclApplicationCommonName: ocscalplugin
userpassword: welcome1

# grant proxy privilege to the application entity
dn: cn=UserProxyPrivilege,cn=Calendar,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
changetype: modify
add: uniquemember
uniquemember: orclApplicationCommonName=ocsCalPlugin, cn=oses, cn=Products, cn=OracleContext
```

csPlugin.ldif

```
# create the application entity
dn: orclApplicationCommonName=ocsCsPlugin,cn=ifs,cn=Products,cn=OracleContext
changetype: add
objectClass: orclApplicationEntity
objectClass: top
orclApplicationCommonName: ocsCsPlugin
userpassword: welcome1

# add the application entity into the uniquemember of the trusted application
dn: cn=Trusted Applications,cn=Groups,cn=OracleContext
changetype: modify
add: uniquemember
```

```
uniquemember: orclApplicationCommonName=ocsCsPlugin,cn=ifs,cn=Products,cn=OracleContext

# add the application entity into uniquemember of the userproxyprivilage
dn: cn=userproxyprivilege,cn=groups,cn=oraclecontext
changetype: modify
add: uniquemember
uniquemember: orclApplicationCommonName=ocsCsPlugin,cn=ifs,cn=Products,cn=OracleContext

# add trusted applications as the application entity's orcltrustedapplicationgroup member
dn: orclApplicationCommonName=ocsCsPlugin,cn=IFS,cn=Products,cn=OracleContext
changetype: modify
add: orcltrustedapplicationgroup
orcltrustedapplicationgroup: cn=Trusted Applications,cn=groups,cn=oraclecontext

# enable Userpassword Reversible Encryptio
dn: cn=PwdPolicyEntry,cn=Common,cn=Products,cn=OracleContext
changetype: modify
replace: orclpwdencryptionenable
orclpwdencryptionenable: 1
```

E

Third Party Licenses

This appendix includes the third party license for all the third party products included with Oracle Secure Enterprise Search.

Apache log4j and Apache Axis

This program contains code from the Apache Software Foundation ("Apache"). Under the terms of the Apache license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache software is provided by Oracle "AS IS" and without any warranty or support of any kind from Oracle or Apache.

The Apache Software License

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.
```

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Glossary

crawl

The process of reading sources and creating the search engine index.

crawler

An Oracle Secure Enterprise Search program that reads sources to create the search engine index.

federated search

A search that is performed by an external entity. It provides a unified framework to search the different document repositories that are crawled, indexed, and maintained separately. A single search can be run across all indexes, and it aggregates the search results to show one result list to the user. User credentials are passed along with the search so that each remote index can authenticate the user against its own document repository.

hitlist

A list of results for a search.

index

An Oracle Secure Enterprise Search structure that is updated after a crawl. It is used to improve performance of searches.

Oracle Secure Enterprise Search administration tool

A tool to manage the search engine, including sources and schedules.

Oracle Secure Enterprise Search application

Application for searching the Oracle Secure Enterprise Search index.

relevance

The level of match of the search results to the search string.

schedule

The frequency with which each source is crawled.

search

The process of querying the search engine.

searchctl

A tool for starting and stopping the search engine.

search metadata

Information about the sources, crawls, and schedules.

secure search

A type of search that only returns results that the user is allowed to view based on access privileges.

seed URL

The starting point for a crawl.

sources

A source of data to be searched. Sources can be Web sites, database tables, files, e-mail, mailing lists, OracleAS Portal page groups, federated sources, Oracle Calendar repositories, Oracle Content Database repositories, or user-defined sources.

Index

G

H

I

J

L

M

O

P

Q

R

S

T

U

W