Oracle® Secure Enterprise Search

Administrator's Guide

10*g* Release 1 (10.1.8)

B32259-01

November 2006

ORACLE®

Oracle Secure Enterprise Search Administrator's Guide, 10*g* Release 1 (10.1.8)

B32259-01

Primary Author: Michele Cyran

Contributors: 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 Configuring Access to Enterprise Content Sources

6 Oracle Secure Enterprise Search Advanced Information

7 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 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 (SES) 10g Release 1 (10.1.8). It also provides pointers to additional information.

New Features in Oracle Secure Enterprise Search Release 10.1.8

The main driver of growth in the enterprise search market: People want a single point of access to all their information.

- Out-of-the-box, with no additional coding required, Oracle SES 10.1.8 provides more access than any other enterprise search engine. It can find and verify information in the following:
  - Files in Microsoft NT File systems (NTFS)
  - EMC Documentum Content Server DocBases
  - IBM Lotus Notes databases
  - FileNet Content Engine object stores
  - FileNet Image Services libraries
  - Open Text Livelink
  - Microsoft Exchange

  Oracle SES ships with *plug-ins* (a plug-in is a software module that adds features by Oracle SES) for all these applications. (Note: To use some of the new plug-ins, additional licensing is required.) Oracle SES controls access to private documents and restricts access to specific workgroups based on access control information obtained during the indexing and stored in its search engine index.

  See Also: Chapter 5, "Configuring Access to Enterprise Content Sources"

- Oracle SES also searches across a number of Oracle sources: OracleAS Portal, Oracle Collaboration Suite Content Services and Calendar, Oracle Content Database, selected modules of Oracle E-Business Suite, and Oracle Siebel.

  See Also: Chapter 5, "Configuring Access to Enterprise Content Sources"

- Oracle SES is now directly integrated with access control and identity management solutions. No synchronization with Oracle Internet Directory is necessary for Oracle SES to ensure access control. Oracle SES can directly access

Active Directory (no extra coding required) through a new authorization API and identity plug-in architecture. Oracle SES ships plug-ins for Oracle Internet Directory and Microsoft Active Directory, among others.

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

- New suggested content feature lets you index and display real time content in the search results screen. A style sheet can be applied to the content before it is displayed in the search result list.

  See Also: "Add Suggested Links or Suggested Content" on page 6-8

- In addition to the existing Query Web Service API, Oracle SES now includes an Admin Web Service API. This API lets you perform a subset of administrative actions, such as starting and stopping a crawler schedule or getting the index fragmentation level. The Admin Web service is located at the following URL: `http://host:port/search/ws/admin/SearchAdmin`.

  See Also:

  Oracle Secure Enterprise Search Java API Reference

  Appendix D, "WSDL Specifications"

  The "Web Services Interface" section in the Oracle SES administration tutorial:

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

- Other improvements include a simplified method for configuring secure search with OracleAS Single Sign-On, a *title fallback* feature to override default document titles picked up during crawling with a more meaningful title later, a more simple configuration of federated sources, and case-insensitive relevancy boosting (documents with "Oracle" are boosted when you enter "oracle".)

  See Also:

  "Title Fallback" on page 3-5

  "Tips for Using Federated Sources" on page 6-3 and "Setting Up Secure Federated Sources" on page 5-37

  Configuring Secure Search with OracleAS Single Sign-On on page 4-11

- Upgrade from Oracle SES Release 1 (10.1.6) is supported.

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 numerous 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 lets you search across multiple OracleAS Portal repositories, such as Web pages, files on disk, and pages on other OracleAS Portal instances.
- **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.

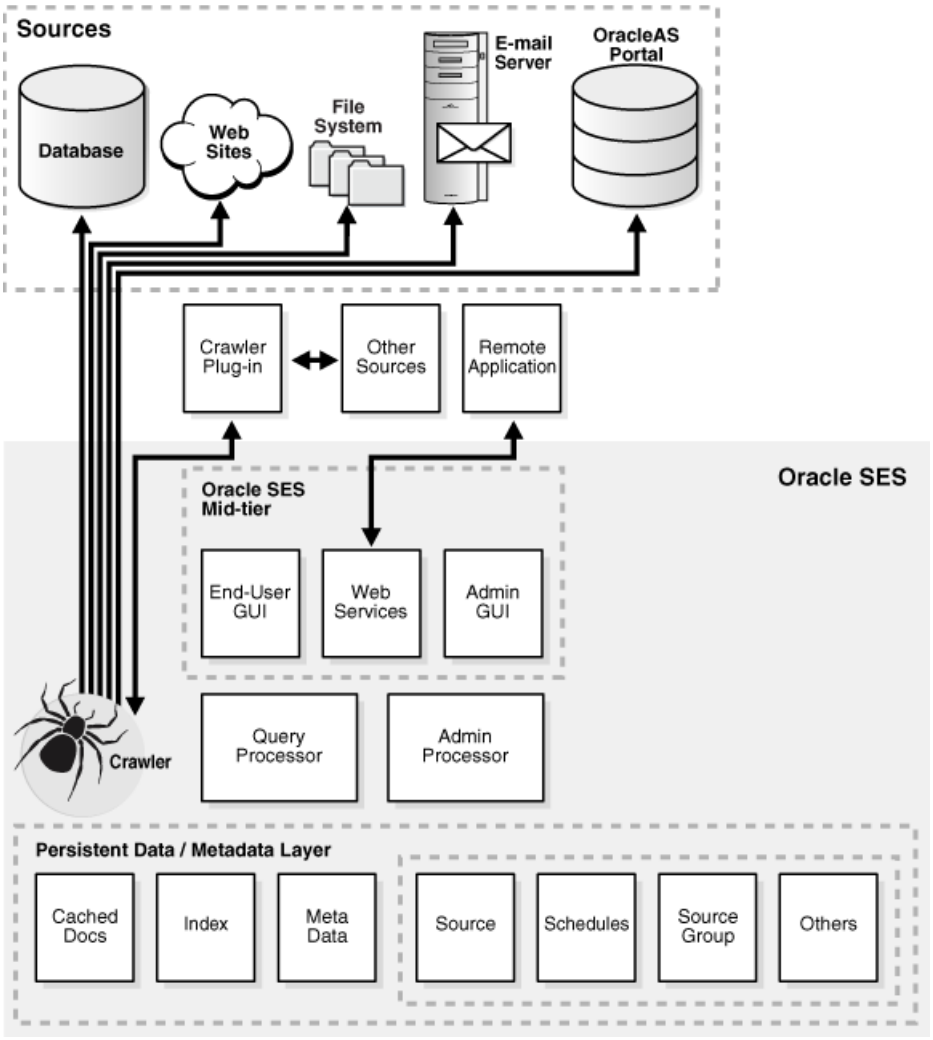
> **Note:** Oracle Content Database and Oracle Content Services are the same product. This book uses the product name Oracle Content Database to mean Oracle Content Database *and* Oracle Content Services. Oracle Content Database sources are certified with Oracle Content Database release 10.2 and Oracle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

- **Oracle Applications (Oracle E-Business Suite 11i and Siebel 8)**: Search Oracle Applications with an Oracle E-Business Suite 11i source or a Siebel 8 source.
- **Federated**: A federated source lets you search secure content across distributed Oracle SES instances.

Additionally, out-of-the-box, with no additional coding required, Oracle SES 10.1.8 provides more access than any other enterprise search engine. It can find and verify information in the following:

- Files in Microsoft NT file systems (NTFS)
- EMC Documentum Content Server DocBases
- IMB Lotus Notes databases
- FileNet Content Engine object stores
- FileNet Image Services libraries
- Open Text Livelink
- Microsoft Exchange

> **See Also:** Chapter 5, "Configuring Access to Enterprise Content Sources"



See Also:

- *Oracle Secure Enterprise Search Release Notes* for version information and known issues
- *Oracle Secure Enterprise Search Installation and Upgrade Guide* 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 administration 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 7, "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 in 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 4-8

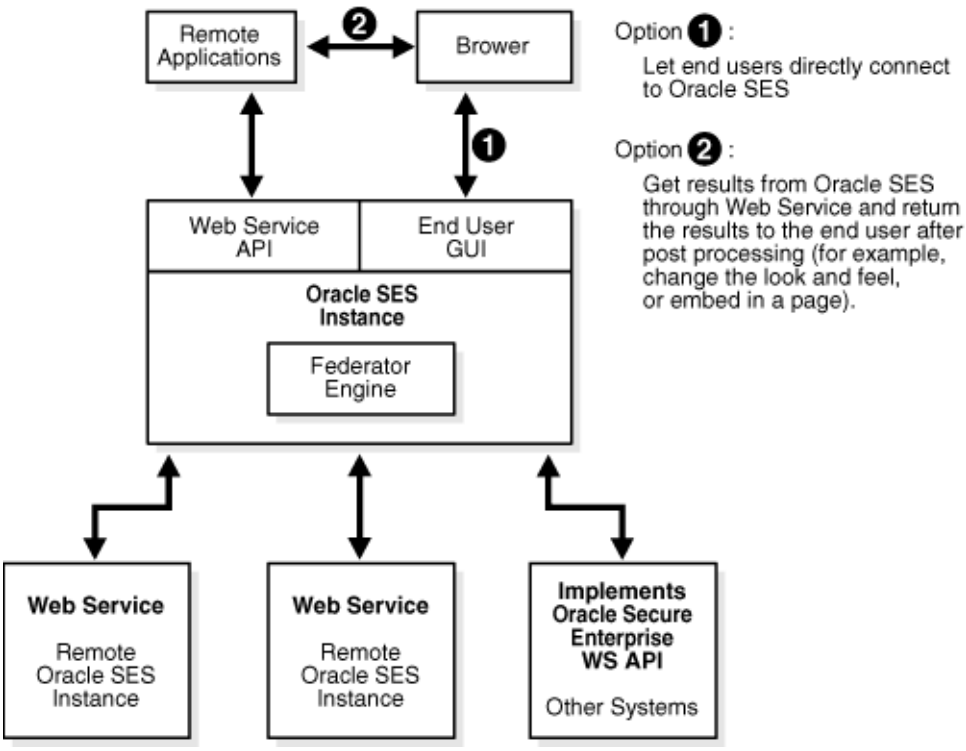
Federated Search

Oracle Secure Enterprise Search 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. A *federation broker* calls the *federation endpoint* to collect content matching the search criteria for the sources managed at that endpoint.

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 federation endpoint can authenticate the user against its own document repository.

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

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 APIs" on page 7-2

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 7-27
> - 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**

  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**

  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 list includes any available user-defined source types. You can create as many sources as you want.

- **Home - Schedules**

  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**

  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**, **Suggested Content**, **Alternate Words**, and **Source Groups** subtabs. These pages help you improve search performance.

- **Search - Relevancy**

  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**

  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 - Suggested Content**

  Suggest actual content (as opposed to links) to be displayed in the result list. For example, when an end-user searches for contact information on a coworker, Oracle SES fetches the content from the suggested content provider and returns the contact information (e-mail address, phone number, and so on) for that person in the result list. Suggested content results appear under any suggested links and above the query results.

- **Search - Alternate Words**

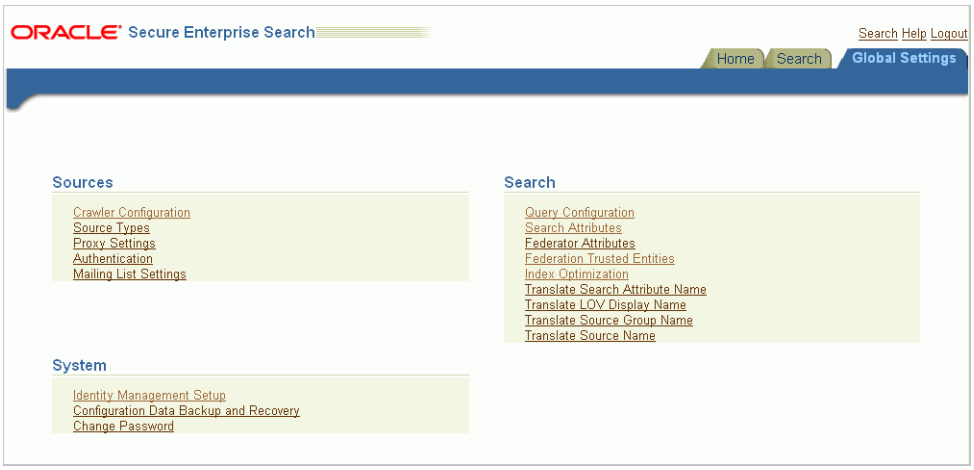
  Use alternate words to suggest alternative search queries to users. This is useful for fixing common errors that users make in searching (for example, entering Oracel instead of Oracle). Also, synonyms can provide more relevant results; for example, cellular phones for cell phones or wireless phones.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

- **Search - Source Groups**

  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**

  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**

  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 users submit URLs to be crawled and indexed. You can examine submitted URLs before they are indexed by the crawler.
  - Federated search - This lets users search secure content across distributed Oracle SES instances.
  - Secure search configuration - This includes options for identity-based security filters (using users and groups from an identity management system) and options for end user authentication. For example:
    * Require login to search secure content. This is the default. Users can search public content without logging in but must login to retrieve secure content.

* Require login to search secure *and* public content. Users must first login to retrieve any content. This option requires that an identity plug-in is activated.

- **Identity Management Setup**

  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 administration 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:

See Also:

- "Tuning Crawl Performance" on page 6-4 and "Tuning Search Performance" on page 6-8
- 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. 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 7-27

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 6-4 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 B, "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 that 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 in 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 7-31
- *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 re-crawl. This means if there is no change to the document, then you must force the change by setting the re-crawl 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`.

Notes:

- When a title is null, Oracle SES automatically indexes the fallback title for all binary documents (for example, .doc, .ppt, .pdf). For HTML and documents, Oracle SES does *not* automatically index the fallback title. This means that the replaced title on HTML or text documents cannot be searched with the title attribute on the **Advanced Search** page. You can turn on indexing for HTML and text documents in the `crawler.dat` file. (For example, set `NULL_TITLE_FALLBACK_INDEX ALL`)
- The `crawler.dat` file is not included in the backup available on the **Global Settings - Configuration Data Backup and Recovery** page. Make sure you manually back up the `crawler.dat` file.

See Also: "Replacing Default Document Titles Using Title Fallback" 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 that 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 6-4

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

Crawler Configuration File

The crawler configuration file is `$ORACLE_HOME/search/data/config/crawler.dat`. All crawler configuration tasks *except* title fallback are controlled in the Oracle SES administration tool. The only reason to configure this file is to replace default document titles using the title fallback feature.

> **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. 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 INFO    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 B, "URL Crawler Status Codes"

Replacing Default Document Titles Using Title Fallback 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 in UTF8 character set.

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 6-8

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.

Table 3–1 (Cont.) Search String Rules

Rule	Description
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.
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

Oracle SES supports the `STRING`, `NUMBER`, and `DATE` (mm/dd/yyyy) attributes with the following operators:

- `CONTAINS` operator applies only to the `STRING` attribute; Oracle SES returns documents with an attribute containing the query terms.
- `EQUALS` operator applies to all three attributes; Oracle SES returns documents with an attribute equaling the query with case-insensitivity.
- `GREATERTHAN` operator applies to `NUMBER` and `DATE` attributes; Oracle SES returns documents with an attribute value greater than or later than the query value.

- LESSTHAN operator applies to NUMBER and DATE attributes.
- GREATERTHANEQUALS operator applies to NUMBER and DATE attributes.
- LESSTHANEQUALS operator applies to NUMBER and DATE attributes.

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 in 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 topics:

About Oracle Secure Enterprise Search Security

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

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.

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

- 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.
- 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.

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.

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: 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 registered identity plug-ins for Oracle Internet Directory and other identity management systems.) 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.

  See Also: "Activating an Identity Plug-in" on page 4-5

- 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 the users or groups of users that 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 authorization).

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 authorization. Extra work is required to implement a `QueryTimeFilter`.

See Also:

- "Admin-based Authorization" on page 4-8 for more information about using ACLs
- "Query-time Authorization" on page 4-9 for more information on using a Java filter class
- "Crawler Plug-in API" on page 7-27

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.

Activating an Identity Plug-in

Activate an identity plug-in on the **Global Settings - Identity Management Setup** page. From the available identity plug-ins, select the one you want to use for authentication and validation activity in Oracle SES, and click **Activate**. For example:

1. For the Active Directory identity plug-in enter values for the following parameters:
   - **Directory URL**: ldap://<*Active Directory server*>:389
   - **Directory account name**: <*User Logon Name*> Confirm the user logon name on the Active Directory Users and Computers application. Under the **User** folder, right-click **username**. Select **Property** and go to the **Account** tab. For example, assume the user account `adtest` in domain `domain1.company.com`, which is associated with the target Active Directory. You may try `domain1\adtest` or `adtest@domain1.company.com` or `cn=adtest, cn=users, dc=domain1, dc=company, dc=com` if you are not sure the actual user logon name. The user account does not need to be an administrator account.
   - **Directory account password**: <*Password for this Directory account*>
   - **Directory subscriber**: dc=domain1,dc=company,dc=com, if your domain name is domain1.company.com
   - **Directory security protocol**: none
2. Click **Finish**.

Re-registering Preinstalled Identity Plug-ins

If a pre-installed identity plug-in is accidentally removed, you can re-register it with the following steps:

1. On the **Global Settings - Identity Management Setup** page, click **Register new Identity Plug-in**.
2. Enter the class name and jar file name of the removed identity plug-in:

Table 4–2 Identity Plug-in Class Names and Jar File Names

Identity Plug-in	Plug-in Class Name	Jar File Name
EMC Documentum Content Services	oracle.search.plugin.security.identity.dcs.DCSIdentityPluginManager	../dcs/DCSIdentityPlugin.jar
FileNet Image Services	oracle.search.plugin.security.identity.fnis.FNISIdentityPluginManager	../fnetis/FNISIdentityPlugin.jar

Table 4–2 (Cont.) Identity Plug-in Class Names and Jar File Names

Identity Plug-in	Plug-in Class Name	Jar File Name
Open Text Livelink	oracle.search.plugin.security.identity.llcs.LLCSIdentityPluginManager	../llcs/LLCSIdentityPlugin.jar
Oracle E-Business Suite 11i	oracle.search.plugin.security.identity.ebs.EBS11IdentityPluginMgr	../oracleapplications/EBS11Crawler.jar
Siebel 8	oracle.search.plugin.security.identity.siebel.SiebelIdentityPluginMgr	../oracleapplications/Siebel8Crawler.jar
Oracle Internet Directory	oracle.search.plugin.security.identity.oid.OIDPluginManager	OIDPlugins.jar
Active Directory	oracle.search.plugin.security.idm.IdentityPluginManagerADImpl	idm/idmPlugin.jar
Lotus Notes	oracle.search.plugin.security.identity.ln.LNIdentityPluginManager	ln/LNIdentityPlugin.jar

3. Click **Finish**.

Restrictions on Changing the Identity Plug-in

The information Oracle SES saves from the identity plug-in (that is, the correspondence between names and canonical attribute values) may not be valid on different identity plug-ins. If you keep the same identity plug-in 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 identity plug-in anytime without restriction. This section describes steps you must perform if you change identity plug-in 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 use a different identity plug-in server, then you must clear the ACL data before deactivating the original identity plug-in. If the ACL data is not cleared, then the ACL policy configured for that source while connected to the old identity plug-in server will not be correctly enforced after connecting to the new identity plug-in server.

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

- Before deactivating the identity 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 identity plug-in 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 identity plug-in, delete each source that uses the ACL policy **Oracle Secure Enterprise Search ACL**. After connecting to the new identity plug-in 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 use a different identity plug-in server, then after activating the new identity plug-in 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 identity plug-in server. Set the **Process All Documents** option on the **Home - Schedules - Edit Schedule** page.

Note: if the ACL data is not cleared before switching identity plug-in 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

The Oracle SES front-end interface collects user credentials, which are then validated against the active identity plug-in. 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. (Oracle SES provides registered identity plug-ins for Oracle Internet Directory and other identity management systems.) 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. It is not included in Oracle SES. 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.

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 in 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.

Oracle SES can use the following two security modes (SSO or non-SSO). This is set on the **Global Settings - Query Configuration** page:

- Require login to search secure content only: anyone can search public content. This is the default. This is also known as secure mode 2.
- Require login to search secure *and* public content. This is also known as secure mode 3.

This is applied to both the default query application and Oracle SES Web services. In mode 3, if a user tries to perform any Web services operation (search or document service) without logging in first, then a SOAP exception is thrown indicating that this secure mode requires login for any operation.

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 administration 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

Within the Crawler Plug-in API, the `DocumentAcl` object implements *identity-based security*. Identity-based security is a security policy based on users and groups that is defined by the active identity plug-in.

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 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.

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 7-27

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 may 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 7-33 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.

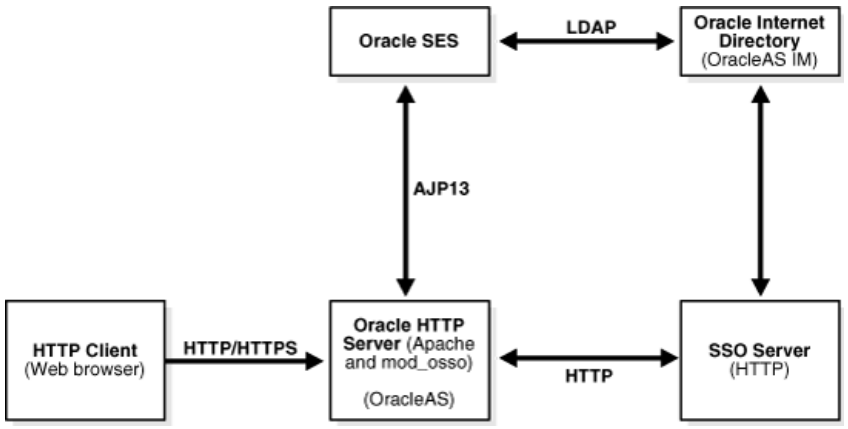
Configuring Secure Search with OracleAS Single Sign-On

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:



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 with SSO. Add the following lines to `$AS_HOME/Apache/Apache/conf/mod_osso.conf` in the `IfModule` element:

```
<Location /search/query/formlogin.uix>
  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 an identity plug-in on the **Global Settings - Identity Management Setup** page.
2. Specify when end-user authentication is required. Oracle SES requires end users to login to search secure content. This is the default. If you want to require end users to login to search both private *and* public content, then change the setting on the **Global Settings - Query Configuration** page.

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. To the `<web-site>` element, add the attribute `protocol="ajp13"`. For example:

```
<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. In 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.

> **Note:** It is important to activate the identity plug-in before you configure SSO. After the Oracle SES instance is behind SSO, identity plug-in activation does not work.

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 4-19.

      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 in 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 Sources" on page 5-37.

5

Configuring Access to Enterprise Content Sources

This chapter contains the following topics:

Introduction to Enterprise Content Sources

Consumer search engines, like Google and Yahoo, search HTML pages. An enterprise search engine, however, must also search databases, e-mail systems, intranet portals, document management systems, and custom applications. Oracle SES ships plug-ins to the most popular of these systems in use today.

Some of the plug-ins shipped with Oracle SES require extra licensing fees. Contact Oracle sales for details.

Individual client libraries may need to be installed (and licensed from the vendor) for some content sources to work. For example, EMC Documentum requires a compatible version of Documentum Foundation Classes (DFC), a Java library, to be installed on the machine running Oracle SES. Oracle SES does not ship with DFC.

> **See Also:** *Oracle Secure Enterprise Search Release Notes* for a list of supported platforms

Identity Management with Enterprise Content Sources

Oracle SES lets you register an identity plug-in as an interface to any identity management system. Oracle SES provides registered identity plug-ins for Oracle Internet Directory, Active Directory, and other identity management systems. 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.

> **See Also:** "Authorization and Authentication" on page 4-2 for information about identity plug-ins

The following table lists which identity plug-ins are available for each enterprise content source.

Table 5–1 Identity Plug-ins for Enterprise Content Sources

Source Type	Versions Supported	Identity Plug-in
EMC Documentum Content Server	5.3 SP2	Active Directory, Oracle Internet Directory, Native
FileNet Content Engine	3.5	Active Directory
FileNet Image Services	4.0 (ISRA 3.2)	Active Directory, Oracle Internet Directory, Native
Lotus Notes	5.0.9, 6.5.4,7.0	Active Directory, Oracle Internet Directory, Native
NTFS	Windows 2000, Windows 2003	Active Directory
Open Text Livelink	9.2, 9.5, 9.5.5	Active Directory, Native
Oracle Calendar	10.1.2 or later	Oracle Internet Directory
Oracle Content Database	Oracle Content Services 10.1.2 or later, Oracle Content Database 10.2	Native, Query-time authorization
Oracle E-Business Suite 11i	11i	Native
Siebel 8	8	Native
Microsoft Exchange	Windows 2000, Windows 2003	Active Directory

> **Tip:** "Re-registering Preinstalled Identity Plug-ins" on page 4-5 for a list of identity plug-ins native to enterprise content sources

Setting Up Secure EMC Documentum Content Server Sources

Documentum data is stored in DocBases, which can contain cabinets and folders. A Documentum Content Server instance can have one or more DocBases crawled with an EMC Documentum Content Server source. The Documentum Content Server source navigates through the DocBases and the inline cabinets to crawl all the documents in Documentum Content Sever. Oracle SES creates an index, stores the metadata, and accesses information in Oracle SES to provide search according to the end user permissions.

Oracle SES supports incremental crawling; that is, it crawls and indexes only those documents that have changed after the most recent crawling was scheduled. A document is re-crawled if either the content or metadata or the direct security access information of the document has changed. A document is also re-crawled if it is moved within Documentum Content Server and the end user has to access the same

document with a different URL. Documents deleted from a DocBase will be removed from the index during incremental crawling.

Important Notes for EMC Documentum Content Server Sources

The admin account of a DocBase should be used by the Documentum source in Oracle SES for crawling and indexing documents of that DocBase.

Required Software

- Documentum Content Server DA (Documentum Administrator) *or* Documentum Content Server WebTop application must be installed and configured.
- Documentum Foundation Classes (DFC) must be installed on the server running Oracle SES.

Required Tasks

- Because EMC Documentum Content Server software is not included with Oracle SES, certain files must be copied manually into Oracle SES.

  The DFC installation asks for destination directory and user directory. For Windows, the default destination directory is `C:\Program Files\Documentum` and default user directory is `C:\Documentum`. For UNIX, it is a prerequisite to create DFC program root and DFC user root. For example, DFC program root can be `<USER HOME>/documentum_shared` and DFC user root can be `<USER HOME>/documentum`.

  Copy the `dfc.properties` and DFC jar files from the following locations into `ORACLE_HOME/search/lib/plugins/dcs`.

  - `dfc.jar`
    - Windows: `<DFC destination directory>/shared/`
    - UNIX: `<DFC destination directory>/dfc`
  - `dfcbase.jar`
    - Windows: `<DFC destination directory>/shared/`
    - UNIX: `<DFC destination directory>/dfc`
  - `dfc.properties`
    - Windows: `<DFC user directory>/config/`
    - UNIX: `<DFC user directory>/config/`

  For Windows 2003 Server, copy `dmcl40.dll` from `<DFC destination directory>/shared/` to `ORACLE_HOME/bin`.

  For UNIX, copy `libdmcl40.so` from `<DFC destination directory>/dfc` to `ORACLE_HOME/lib`.

  > **Note:** The environment variable `$DOCUMENTUM_SHARED` (DFC Program root) and `$DOCUMENTUM` (DFC user directory) must be created before installing DFC on UNIX. See the DFC installation guide for more information.

- Push the DCS libraries to global libraries by adding the following lines to the `oc4j/j2ee/OC4J_SEARCH/config/application.xml` file:

```
<library path="../../../../search/lib/plugins/dcs/dfcbase.jar" />
<library path="../../../../search/lib/plugins/dcs/dfc.jar" />
<library path="../../../../search/lib/plugins/dcs" />
<library path="../../../../search/lib/log4j.jar" />
```

This assumes that the directory `search/lib/plugins/dcs` contains the Documentum Server configuration file `dfc.properties`.

Known Limitations

In this release, search results cannot be viewed in Documentum desktop. The documents and folders can be viewed only using Documentum Administrator (DA) or Webtop applications.

Setting Up Identity Management for EMC Documentum Content Server

Activate the identity plug-in on the **Global Settings - Identity Management Setup** page. Select Oracle Internet Directory identity plug-in and click **Activate**.

Enter values for the following parameters:

- For **Authentication Attribute**, select **nickname**.
- For **Host name**, enter the host name of the machine where Oracle Internet Directory is running.
- For **Port**, enter the value 389 (the default LDAP port number).
- For **Use SSL**, enter true or false.
- For **Realm**, enter the Oracle Internet Directory realm; for example, dc=us,dc=oracle,dc=com.
- For **User name**, enter the Oracle Internet Directory Administrator user name; for example, cn=orcladmin.
- For **Password**, enter the password for the user name.

Compatible version of Documentum Foundation Classes (DFC) must be installed on the machine running Oracle SES.

1. Import users/groups from Oracle Internet Directory to Documentum. First, create an LDAP Configuration Object in Documentum Administrator (DA):
   a. Login to DA.
   b. Navigate to **Administration - User Management - LDAP**.
   c. Click **File - New - LDAP Configuration Object**.
   d. For **Name**, enter a name for the ldap configuration object.
   e. For **User Subtype**, select **dm_user**.
   f. For **Communication Mode**, select **Regular**.
   g. For **Import**, select **Users and Groups**.
   h. Use this configuration object in the server field select **Default Configuration Object**.
   i. Click **Next**.
   j. For **Directory Type**, select **Oracle Internet Directory Server**.
   k. For **Bind Type**, select **Bind by Searching for Distinguished Name**.

l. For **Binding Name**, enter the Administrator user name of Oracle Internet Directory, normally cn=orcladmin.

m. For **Binding Password**, enter the Administrator password of Oracle Internet Director.

n. For **Host Name**, enter the Oracle Internet Directory host name.

o. For **Port**, it shows the default value 389 (the default port number of Oracle Internet Directory).

p. For **Person Object Class**, enter the information of Base Person Object, typically the value is inetOrgPerson.

q. For **Person Search Base**, enter the person search base defined in Oracle Internet Directory; for example, dc=Users,dc=us,dc=oracle,dc=com.

r. For **Person Search Filter**, specify the cn=*.

s. For **Group Object Class**, enter the Group Object; typically, its value is groupOfUniqueNames.

t. For **Group Search Base**, enter the Group Search base defined in Oracle Internet Directory; for example, cn=Groups,dc=us,dc=oracle,dc=com.

u. For **Group Search Filter**, specify the cn=*.

v. Click **Next**.

w. Attribute Map information is displayed. Click **Finish**.

2. Run the LDAP_Synchronization job:

a. Login to DA.

b. Navigate to **Administration - Job Management - Jobs**.

c. Open the job **dm_LDAPsynchronization**.

d. For **state**, select **Active**.

e. Check the **Deactivate On Failure** check box.

f. For **Designated Server**, select the host name of Documentum Server.

g. Check the **Run After Update** check box.

h. Go to the **Schedule** tab.

i. For **Start Date And Time**, set the current date and time.

j. Select **Repeat time** from the **Repeat** list.

k. Set **Frequency** to any numeric value.

l. Select the **End Date And Time** radio button and specify how long the synchronization job should run.

m. Go to the **Method** tab.

n. Check the **Pass Standard Argument** check box.

o. Go to the **SysObject info** tab.

p. Click **OK**.

3. Add permission to each folder and file to make them accessible by the search user. (Adding permissions to a folder automatically adds the same permissions to all files and sub-folders in the folder.) The following steps create a permission set and

assign users/groups to that set. The same permission is assigned to documents. If the documents are not stamped with permission, then it won't get crawled.

Create Access Control Lists (ACLs):

a. Login to DA.

b. Navigate to **Administration - Security**.

c. In the **File** menu click **File - New - Permission set**.

d. For **Name**, enter a name for the permission set.

e. Click **Next**.

f. Click **Add** to add more users/groups to the permission set.

g. Select LDAP users/groups that are to made a part of the permission set and move them to the right frame using the arrow keys. Click **OK**.

h. Click **Finish**.

4. Assign ACLs to documents:

   a. Login to DA.

   b. Navigate to the document where the permission set is to be applied.

   c. Select the **Properties** icon of this document.

   d. Go to the **Permissions** tab.

   e. Click **Select** in front of **Permission set name**.

   f. Search and select the permission set to be applied to the document.

   g. Click **OK**.

It is important that the users/groups in the permission sets that are applied to the documents are LDAP users/groups. Those documents that do not have permission sets with LDAP users/groups will not be crawled.

Creating an EMC Documentum Content Server Source

Create an EMC Documentum Content Server source on the **Home - Sources** page. Select EMC Documentum Content Server from the Source Type list, and click **Create**. Enter values for the following parameters:

- **User name:** Enter the user name of a valid Documentum Content Server user. The user should be an administrator user or a user who has access to all cabinets/folders and documents of the DocBases configured in the **Container name** parameter. The user should be able to retrieve content, metadata, and ACL from cabinets, folders, documents and other custom sub classes of all DocBases configured in **Container name** parameter. This is a required parameter.
- **Password**: Password of the Documentum user. This is a required parameter.
- **Container name**: The names of the containers to be crawled by Oracle SES. You can crawl an entire Documentum DocBase or a specific cabinet/folder. The format is <DocBase Name>/<Cabinet Name>/<Folder Name>/<Sub Folder Name>. Multiple comma-delimited container names can be entered. This parameter is case-sensitive; hence, the same cabinet name as in Documentum repository should be entered. This is a required parameter. For example:
  - Container name: DocBase1: The entire DocBase1 will be crawled.

  - Container name: DocBase2/Cabinet21: Cabinet21 and its sub-folders within DocBase2 will be crawled.
  - Container name: DocBase2/Cabinet21/Folder11: Folder11 and its sub-folders will be crawled.
  - Container name: DocBase1, DocBase2/Cabinet21/Folder11: The entire DocBase1 and Folder 11 in DocBase2/Cabinet21 will be crawled.

- **Crawl folder attributes**: Indicate whether folder attributes need to be crawled, either true or false. This is an optional parameter. The default value is false. If any other value is provided, it is assumed to be false.
- **Crawl versions**: Indicate whether multiple versions of documents should be crawled, either true or false. This is an optional parameter. The default value is false. If any other value is provided, it is assumed to be false and only the latest versions of a document will be crawled.
- **Attribute list**: The comma-delimited list of Documentum attributes along with their data types to be searchable. The format is <Attribute Name>:<Attribute Type>, <Attribute Name:Attribute Type>. Valid values are String, Number, and Date.

Table 5–2 Documentum Data Type Mapping

Sr. No	Documentum Data Type	Oracle SES Data Type
1	Boolean	Number
2	Integer	Number
3	String	String
4	ID	String
5	Time or Date	Date
6	Double	Number

While crawling a DocBase, an attribute is indexed only if both name and type match the configured name and type; otherwise, it is ignored. This is an optional parameter. For example: To make the following Documentum attributes searchable:

  - Attribute Name: Account Name Attribute Type: String
  - Attribute Name: Account Id Attribute Type: Integer
  - Attribute Name: Creation Date Attribute Type: Date

The value of **Attribute list** should be the following:

Account Name: String, Account ID: Number, Creation Date:Date

The default searchable attributes for Documentum Content Server are Modified Date, Title, and Author.

Multiple attributes with same name are not allowed. For example, Emp_ID:String, Emp_ID:Number

- **URL for Viewing the Document**: A valid URL for Documentum WebTop or DA application used for viewing the Oracle SES search results. For example, http://<IP address>:<Port No>/da or http://<IP address>:<Port No>/webtop.
- **Authentication Attribute**: This parameter is used to set ACLs. This parameter lets you set multiple LDAP servers. If Oracle SES and Documentum Content Server

are synchronized with Active Directory, then enter the value `USER_NAME`. If Oracle Internet Directory is used, then enter `nickname`.

Setting Up Secure FileNet Content Engine Sources

FileNet Content Engine data is stored in object stores, which can be further contained inside folders on a server. A FileNet Content Engine instance can have one or more object stores that can be crawled by specifying the Object Store details in the **Container name** parameter in Oracle SES. The Content Engine source navigates the object store to crawl all the documents in the configured Content Engine Object Store. It stores the metadata and accesses information in Oracle SES to provide search according to the end user permissions.

Important Notes for FileNet Content Engine Sources

Any user having administrative privileges can be used to access FileNet Content Engine Crawler plug-in for crawling and indexing documents.

Required Software

- FileNet Content Engine version 3.5
- FileNet Application Engine version 3.5

Required Tasks

Because FileNet Content Engine software is not included with Oracle SES, certain files must be copied manually into Oracle SES:

- Copy `javaapi.jar`, `soap.jar`, `xercesImpl.jar` and `xml-apis.jar` files from `<FileNet installed Folder>/Workplace/WEB-INF/lib` to `ORACLE_HOME/search/lib/plugins/fnetce`.
- Copy the `WCMConfig.properties` file from `<FileNet installed Folder>/Workplace/WEB-INF`, into `ORACLE_HOME/search/lib/plugins/fnetce`.

Known Limitations

- If any of the parameters are updated after initial crawl, then you must update the crawler re-crawl policy to **Process All Documents** on the **Home - Schedules - Edit Schedules** page, and re-crawl the source.
- If additional document types are configured after first time crawl, these document types are not indexed on subsequent re-crawls. Same is the case if Document Size parameter is changed after first crawl, for example if the Document Size was 10 MB at the time of first crawl and it is changed to 20 MB before re-crawl, documents greater than 10 MB are be rejected. Workaround is to create the source again and then make the changes.

Setting Up Identity Management with Filenet Content Engine

If a FileNet Content Engine source is used, Oracle recommends that Active Directory be used as identity management system for the Oracle SES instance. The Active Directory instance must be the same one that FileNet Content Engine is using to authenticate users on the file system.

See Also: "Activating an Identity Plug-in" on page 4-5 for information on activating the Active Directory identity plug-in

Creating a FileNet Content Engine Source

Create a FileNet Content Engine source on the **Home - Sources** page. Select FileNet Content Engine from the Source Type list, and click **Create**. Enter values for the following parameters:

- **User name**: A valid FileNet Content Engine user. The user should be an Administrator user or a user who has access to all Folders and Documents present in the configured container. The user should be able to retrieve content, metadata, and ACL from folders, documents of all containers configured in **Container name** parameter. This is a required parameter.
- **Password**: Password of the Content Engine user. This is a required parameter.
- **Container name**: The names of the containers to be crawled by Oracle SES. You can crawl a complete objectstore or a specific Folder. The format for specifying container is <ObjectStore>/<Folder Name>/<Sub Folder Name>. Multiple comma-delimited containers can be specified. This is a required parameter. For example:
  - Container name: ObjectStore1: The entire ObjectStore1 will be crawled.
  - Container name: ObjectStore1/Folder1/Folder12: The documents inside Folder12 and its sub-folders will be crawled.
  - Container name: ObjectStore1, ObjectStore2/Folder1/Folder12: The entire ObjectStore1 and contents of Folder12 in ObjectStore2 will be crawled.
- **Attribute list**: Attribute list corresponds to the comma-delimited list of Content Engine attributes along with their data types that the administrator wants to be searchable. The format is <Attribute Name>:<Attribute Type>, <Attribute Name:Attribute Type>. The valid values are String, Number, and Date.

Table 5–3 FileNet Content Engine Data Type Mapping

Sr. No	FileNet Content Engine Data Type	Oracle SES Data Type
1	Boolean	String
2	float, int, byte, and other numeric values	Number (Big Decimal)
3	String	String
4	DateTime, Date	Date
5	Others	String

While crawling from object store an attribute will be indexed only if a valid attribute name and data type should be matched with the configured name and type, else it will be ignored. This is an optional parameter. For example, to make the following Content Engine attributes searchable:

- Attribute Name: DocumentTitle Attribute Type: String
- Attribute Name: Id Attribute Type: Number
- Attribute Name: DateCreated Attribute Type: Date

The value of Attribute List should be: Document Title: String, Id: Number, DateCreated: Date

The default searchable attributes for FileNet Content Engine are Title, Author, and Last Modified Date. Multiple attributes with same name are not allowed. For example: Emp_ID: String, Emp_ID: Number is not allowed.

- **Crawl versions**: Indicate multiple versions of documents to be crawled with true. By default, this value is false; that is, only the latest version of documents will be crawled. If any value other than true is specified, it is considered false.
- **URL for viewing the documents**: The URL for FileNet Workplace application used for viewing the search results. Workplace is a part of FileNet P8 AE. For example: http://<IP address> :< Port No.>/Workplace
- **Remove deleted documents from index**: This parameter determines whether documents deleted from CE object stores should be removed from the index as well, either true or false. The default value is false, as this would be a costly operation in terms of performance. If any value other than true is specified, it is considered false.
- **Crawl folder attributes**: Specify whether or not folder metadata should be indexed, either true or false. The default value is false. Any other value for this parameter is considered false.

Setting Up Secure FileNet Image Services Sources

Documents in FileNet Images Services are organized into Folders. A FileNet Image Services source navigates through the folder hierarchy to crawl all documents in FileNet Image Services (IS). Oracle SES creates the index and stores the metadata of the documents retrieved from FileNet Images Services in Oracle SES to provide search according to the end users' permissions.

A FileNet Image Server instance can have one or more Libraries. A Library is the document repository and contains documents within Folders and sub-Folders. A FileNet Image Services source can crawl multiple Libraries.

Images stored in Image Services can have annotations. Some of the annotations contain text, and these annotations will be crawled. The annotations crawled are:

- Stamp
- Transparent Text
- Stick note

You can search on the content of these annotations after the IS library has been crawled.

Important Notes for FileNet Image Services Sources

A user belonging to IS SysAdmin group should be used to crawl documents and metadata in IS.

Required Software

- FileNet Image Services Server version 4.0 or 3.6 SP2
- Image Services Resources Adapter version 3.2.1

Required Tasks

Because FileNet Image Services software is not included with Oracle SES, certain tasks must be performed manually to integrate with Oracle SES:

- Deploy the `ISCrawlerWeb.war` file in the same application server on which ISRA has been deployed.
- For application servers that require context root to be specified while deploying a WAR file, specify Context Root as `ISCrawlerWeb`.
- If the application server is WebSphere Application Server, then activate URL rewriting: **Click Servers - Application Servers - name of the server - Web Container - Session Management - Enable URL Rewriting**.

Known Limitations

- If additional document types are configured after the first crawl, these document types are not indexed on subsequent re-crawls. The same applies if the **Document Size** parameter is changed after first crawl. For example, **Document Size** was 10 MB at the time of first crawl and it is changed to 20 MB before re-crawl, then documents with greater than 10 MB are rejected. As a workaround: update the crawler re-crawl policy to **Process All Documents** on the **Home - Schedules - Edit Schedules** page, and re-crawl the source.
- XML documents are crawled by default without configuring the source for XML documents: Oracle SES provides an option of configuring the documents types, including XML, to be crawled. Currently, even if XML document type is not configured, XML documents still are crawled.

Setting Up Identity Management for FileNet Image Services

Activate the identity plug-in on the **Global Settings - Identity Management Setup** page.

Configure Oracle SES to Active Directory:

1. On the **Global Settings - Identity Management Setup** page, click **Register new Identity Plug-in**.
2. For **Plug-in Manager Class Name**, enter `oracle.search.plugin.security.idm.IdentityPluginManagerADImpl`
3. For **Plug-in Manager Jar File Name**, enter `idm/idmPlugin.jar`.
4. Click **Finish**.
5. Select the radio button for **The Active Directory Identity Plug-in Manager implemented based on Oracle User & Role API** and click **Activate**.
6. For **Authentication Attribute**, select **USER_NAME**.
7. For **Directory URL,** enter the host name and port number; for example, ldap://ldapserverhost:port.
8. For **Directory account name,** enter the Active Directory user; for example, Administrator.
9. For **Directory account password** enter the password of the Active Directory user.
10. For **Directory subscriber**, enter the Active Directory information like Directory subscriber (ldap base) like 'dc=us,dc=oracle,dc=com'.
11. For **Directory security protocol**, enter none or the port number.
12. Click **Finish**.

Configure the identity plug-in for Image Services

1. Go to the **Global Settings - Identity Management Setup page** in Oracle SES.

2. Create a new directory under [oracle_home]/product/[version]/ [SES Instance Name]/search/lib/plugin/Identity/ for example IdentityPlugin_folder.
3. Copy the FileNet Image Services identity plug-in jar to that folder.
4. Click **Register new Identity Plug-in**.
5. For **Plug-in Manager Class Name**, enter oracle.search.plugin.security.identity.fnis.FNISIdentityPluginManager
6. For **Plug-in Manager Jar File Name**, enter identity/fnis/FNISIdentityPlugin.jar.
7. Click **Finish**.
8. Select the radio button for **The Image Services Identity Plug-in Manager implemented based on Oracle User & Role API** and click **Activate**.
9. For **Authentication Attribute**, select NATIVE.
10. For **Web Component URL** enter the host name and port number of the Web component URL; for example, http://webserverhost:port/ISCrawlerWeb.
11. For **Administrator user name**, enter Image Services user name.
12. For **Administrator password**, enter the password of the Image Services user.
13. For **Library name of IS Server**, enter the name of the Image Services library like 'ISCF'. Library Name is the ISRA connection factory name that is created when ISRA is deployed.
14. Click **Finish**.

Image Services Resources Adapter (ISRA) must be deployed on a supported application server. See the ISRA documentation for supported application servers.

Connection Factory must be created for ISRA, the connection factory should be configured for the target IS libraries. See the ISRA documentation for deployment instructions.

ISRA comes with a viewer application for viewing images and annotations, the `FNImageViewer.ear` application should be deployed on the same application server as ISRA. This viewer would be invoked to display images for example jpeg, tiff, bmp, gif, and annotations. See the ISRA documentation for deployment instructions.

To support secure search, the Image Services server must be synchronized with the Active Directory server. See the section 'LDAP configuration' in ISRA deployment guides for importing Microsoft Active Directory users/groups to Image Services.

After Active Directory users/groups have been imported into Image Services, ISRA must be configured to authenticate with Active Directory. See the section 'LDAP configuration' in ISRA deployment guide for details.

Creating a FileNet Image Services Source

Create a FileNet Image Services source on the **Home - Sources** page. Select FileNet Image Services from the Source Type list, and click **Create**. Enter values for the following parameters:

- **User name**: Enter the user name of a valid FileNet Image Services user. The user should be a SysAdmin user or a user who has access to all Folders and Documents of the Libraries configured in the **Container name** parameter. The user should be able to retrieve content, metadata and ACL from folders, documents and other custom sub classes. The user should be defined in the configured LDAP server and should be imported into IS. This is a required parameter.

- **Password**: The FileNet Image Services user password. This is a required parameter.
- **Container name**: The names of the containers to be crawled by Oracle SES. You can crawl an entire FileNet Image Services Library or a specific Folder. The format is <Library Name>/<Folder Name>/<Sub Folder Name>(cache name). Library name is the ISRA connection factory name created when ISRA is deployed. Cache name is in which the document content can be found. Multiple comma-delimited container names can be entered. This is a required parameter. For example:
  - Container Name: LibraryName1(cache name): The entire LibraryName1 will be crawled
  - Container Name: LibraryName2/Folder1/(cache name): Folder1 and its sub-folders will be crawled.
  - Container Name: LibraryName1, LibraryName2/Folder1(cache name): The entire LibraryName1 and Folder 1 in LibraryName2 will be crawled
  - Cache name: The format is cache name:DomainName:Oraganization. This is an optional parameter, if the cache name is not provided the plug-in tries to retrieve document content from the default page cache. However, the plug-in throws an error if an invalid page cache or empty brackets () is specified. Ask IS administrator for cache details.
- **Attribute names**: The comma-delimited list of Image Services attributes along with their data types to search. The format is <Attribute Name> :<Attribute Type>, <Attribute Name: Attribute Type>. Valid values are String, Number, and Date.

Table 5–4 FileNet Image Services Data Type Mapping

Sr. No	FileNet Image Services Data Type	Oracle SES Data Type
1	BOOLEAN	String
2	BYTE	Number
3	UNSBYTE	Number
4	SHORT	Number
5	UNSSHORT	Number
6	LONG	Number
7	UNSLONG	Number
8	ASCII	String
9	TIME	Date
10	DATE	Date
11	MENU	Number
12	FP_NUM	Number

While crawling a Library an attribute will be indexed only if both name and type of the attribute in the library match the configured name and type; otherwise, it is ignored. This is an optional parameter. For example, to make the following FileNet Image Services attributes searchable:

- Attribute Name: Account Name Attribute Type: String
- Attribute Name: Account Id Attribute Type: Integer

  - Attribute Name: Creation Date Attribute Type: Date

  The value of Attribute List should be

  Account Name: String, Account Id: Number, Creation Date: Date

- **Set source hierarchy**: Indicate whether the source should set the source hierarchy of the document, either true or false. The default value is false. If any other value is provided, it is assumed to be false.

  A document in Image Services can be filed in multiple folders, it is possible that a user could have READ permissions on a document but not on all the folders in which the document is filed. If **Set Source Hierarchy** is 'True', then there is a possibility that a user could view a source hierarchy on which he does not have permissions in IS. However, he would not be able to view the documents on which he does not have READ permissions.

- **Web component URL**: The URL of J2EE application server where the crawler plug-in Web component module is deployed. The format of the URL is http://<host name> :< Port Number>. This is a required parameter.

  The Web component is also used to view the search results, on clicking an Oracle SES search result the user is prompted for login. On successful login, the document is displayed. To view images and annotations the FileNet Image viewer `FNImageViewer.ear` should be deployed. `FNImageViewer.ear` is a part of ISRA CD. If the viewer is not deployed, the images will be displayed in native viewer or the user is prompted to download the document.

- **Set public access**: Indicate whether the source should set the public access of the documents whose ACL is Anyone, either true or false. The default value is false. If any other value is provided, it is assumed to be false.

- **Authentication attribute**: This parameter is used to get the LDAP authentication attribute. This parameter will vary based on the identity plug-in used for authentication. For Microsoft Active Directory, it should be `USER_NAME`. For FileNet Image Services identity plug-in, it should be `NATIVE`.

Setting Up Secure Lotus Notes Sources

Lotus Notes data is stored in notes-databases, which can be further contained inside directories on a server. A Lotus Domino Server instance can have one or more databases that can be crawled using the Lotus Notes source. The Lotus Notes source navigates through the databases to crawl all the documents in the specified databases. It stores the metadata, and accesses information in Oracle SES to provide search according to the end users' credentials.

The Lotus Notes source supports incremental crawling; that is, it crawls and indexes only those documents that have changed after recent most crawling was scheduled. A document is re-crawled if either the content, metadata, display URL or the direct security access information of the document has changed. Documents deleted from a database will be removed from the index during incremental crawling.

Important Notes for Lotus Notes Sources

The user-account used to crawl Lotus Notes databases should preferably be an Administrator account, such that it has access on all databases and is able to retrieve and crawl all documents in the specified databases.

Required Software

- Lotus Domino Server R5.0.9/R6.5.4/R7.0
- Notes Clients R5.0.9/R6.5.4/R7.0

Required Tasks

The following tasks must be performed before installing the Lotus Notes source:

1. HTTP and DIIOP tasks must be running on Domino Server.
2. If the Active Directory identity plug-in is used, then the users and user-groups in the Domino Directory must be synchronized with Active Directory. While using the Active Directory identity plug-in, the short-name in the Lotus Notes person document is used for validating the user in Active Directory, so it should be a resolvable logon name in Active Directory.
3. Configure the server document:
   a. Open the server document on the Lotus Notes server that needs to be crawled.
   b. On the **Configuration** page, expand the server section.
   c. On the **Security** page, in the **Programmability Restrictions** area, specify the appropriate security restrictions for your environment in the following fields:

      Run restricted Lotus Script/Java agents

      Run restricted Java/Javascript/COM

      Run unrestricted Java/Javascript/COM

      For example, you might specify an asterisk (*) to allow unrestricted access by Lotus Script/Java agents, and specify user names that are registered in the Domino Directory for the Java/Javascript/COM restrictions.

      Note: The crawler that you configure to crawl this server with the DIIOP protocol must be able to use the user names that you specify in these fields.

   d. Open the **Internet Protocol** page, then open the HTTP page, and set the **Allow HTTP Clients to Browse Database** option to **Yes**.
   e. Configure the user document:

      Open the user document on the Lotus Notes server that needs to be crawled. This document is stored in the Domino directory.

      On the **Basics** page, for **Internet password**, specify a password.

   f. Restart the DIIOP task on the server.

Known Limitations

- A Lotus Notes source does not index encrypt fields, and the content of attachments with encrypted documents, for searching. With encrypted documents, the URL of the search result launches the Notes document in place of the attachment file, which is the case when non-encrypted documents are crawled.
- Oracle SES currently does not support crawling inside specific folders/views of the Notes custom-applications or mail-databases.

- Oracle SES currently launches the search result documents on the Web browser only and does not yet support the launch for Notes thick client.
- A user cannot login through the Oracle SES search page, when working with Lotus Notes Release 6 identity plug-in. However, this scenario works fine when using Active Directory plug-in.
- During source configuration, if you enter multiple attributes with the same name, the crawler considers the first attribute and ignores the others with the same name.

Setting Up Identity Management for Lotus Notes

Activate an identity plug-in on the **Global Settings - Identity Management Setup** page. Select the identity plug-in for Microsoft Active Directory click **Activate**.

The users/groups on Active Directory can be synchronized with Lotus Domino Directory such that all users/groups in Active Directory get registered in Domino as well. Thus, any ACL entry in a notes database or notes document can be validated in Active Directory also, and vice versa.

Oracle SES also provides a Lotus Notes identity plug-in so the Lotus Domino Directory can be used to authenticate and validate the notes native users and groups in Oracle SES. To use the Lotus Notes identity plug-in:

1. Register the Lotus Notes identity plug-in by providing the following parameters:
   - **Plug-in Manager Class Name** = oracle.search.plugin.security.identity.ln.LNIdentityPluginManager
   - **Plug-in Manager Jar File Name** = <lotus notes identity plug-in folder>/ln/LNIdentityPlugin.jar
2. Activate the Lotus Notes identity plug-in with the following parameters:
   - **Server name**: The Domino server fully qualified host name/IP address. If the HTTP port on the Domino server is not 80, then the host name should be "<server-name> :< HTTP port number>".
   - **User name**: Enter user name of a valid Lotus Domino Server user. This is a required parameter.
   - **Password**: Internet password of the Lotus Notes user. This is a required parameter.

Creating a Lotus Notes Source

Create a Lotus Notes source on the **Home - Sources** page. Select **Lotus Notes** from the Source Type list, and click **Create**. Enter values for the following parameters:

- **User name**: Enter the user name of a valid Lotus Domino Server user. The user should be an Administrator user or a user who has access to all Folders and Documents of the databases configured in the **Container name** parameter. The user should be able to retrieve content, metadata, and ACL from documents of all databases configured in **Container name** parameter. This is a required parameter.
- **Password**: Internet password of the Lotus Notes user. This is a required parameter.
- **Container name**: The comma-delimited names of the containers to be crawled by Oracle SES. These containers could be one or many specific databases or Directory-names if all databases in the particular directories need to be crawled. Multiple database or directory names should be separated by a comma. This is a required parameter.

- **Attribute list**: The comma-delimited list of Lotus Notes attributes along with their data types to search. The format is <Attribute Name> :< Attribute Type>, <Attribute Name: Attribute Type>. The valid values are String, Number, and Date.

Table 5–5 Lotus Notes Data Type Mapping

Sr. No	Lotus Notes Data Type	Oracle SES Data Type
1	Boolean	String
2	Integer	Number (Big Decimal)
3	String	String
4	Date	Date

While crawling a database, an attribute is indexed only if both name and type match the configured name and type; otherwise, it is ignored. This is an optional parameter.

The default searchable attributes for Lotus Domino Server are Modified Date, Title, and Author. Multiple attributes with same name are not allowed.

- **Server name**: The Domino server fully qualified host name/IP address. If the HTTP port on the Domino server is not 80, then the host name should be "<server-name> :< HTTP port number>". This is a required parameter.
- **Crawl public documents**: Indicate whether the public documents on notes databases need to be crawled such that they are available to anonymous users in Oracle SES, either true or false. This is a required parameter.
- **Authentication attribute**: The attribute used to validate the ACL. With Active Directory identity plug-in, the value should be `USER_NAME`. With the Lotus Notes identity plug-in, the value should be `NATIVE`. This is a required parameter.
- **Mail template name**: This parameter is specific to the mail-databases and the mail template's name should be specified here if any/all of the databases being crawled are mail databases. This is a mandatory parameter if either the **Past Days** or **Future Days** parameter is specified.
- **Past days**: If the user is crawling calendar entries, then this parameter specifies the number of days in the past for which the calendar entries are picked. The date of reference here is the start date of the event. This accounts for the number of days in the past, and it does not filter the search by time.
- **Future Days**: If the user is crawling calendar entries, then this parameter specifies the number of days in the future for which the calendar entries are picked. The date of reference here is the end date of the event. This accounts for the number of days in the future, and it does not filter the search by time.
- **Notes title**: Because in Lotus Notes custom applications it is not mandatory to maintain a Title field, this parameter has been provided where the administrator can specify those text fields that should be parsed to retrieve the title field. In case of multiple field names, the first field available on the document would be picked for the title. This is a required parameter.

Setting Up Secure NTFS Sources for Windows

This section includes information for Windows NT File System (NTFS) source on Windows. There is a separate source type for NTFS on UNIX.

The NTFS connector enables Oracle SES to search file repositories in Microsoft NTFS. An Oracle SES NTFS source collects the content, metadata attributes and ACLs of files in NTFS. An NTFS source supports incremental crawl. After the initial crawl is performed, subsequent crawls only collect those documents that have changed since the last crawl. A document is re-crawled if the content, metadata, or the ACL information of the document has changed. A file is also re-crawled if it is moved between folders. Files deleted from NTFS are removed from the index during incremental crawls.

Important Notes for NTFS Sources

- The operating system user running the Oracle SES instance must have read permission on the NTFS file share being crawled. For example, if the remote file share `\\machine1\share1\directory1\` is crawled by the NTFS source, then the SES instance must be run as a domain user who has access to the file share.
- If you get the ACL in the form <encrypted acl>@domain for a folder on a remote machine, it probably means that the machine running the Oracle SES instance and the remote machine are on different domains and your machine cannot interpret the ACLs appropriately.

Required Software

- Windows .NET Framework 2.0
- Microsoft Developer Support OLE File Property Reader (dsofile)

Required Tasks

1. If not already installed, download and install the Windows .Net 2.0 Framework:

   See Also:

   `http://msdn.microsoft.com/netframework/downloads/updates/default.aspx`

2. If not already installed, download and install Microsoft Developer Support OLE File Property Reader.
3. Register `dsofile.dll` in the Windows operating system using `regsvr32.exe`.

The Oracle SES process needs to be run as domain administrator to crawl remote machines on the domain This is an important pre-requisite to crawl the remote machines for NTFS. Follow these steps to run Oracle SES process as the domain administrator:

1. Navigate to **Control Panel** - **Administrative Tools** - **Services**.
2. Select the process `OracleService<db sid>`.
3. Stop this process.
4. Right click and select **Properties**.
5. Select the **Log on** tab.
6. Select the option **This account**, and enter the domain administrator name and password.
7. Start this process.

Note: If the Oracle SES instance fails to start after the preceding change, then follow these steps:

1. Navigate to the `$ORACLE_HOME/NETWORK/ADMIN` directory.
2. Edit `sqlnet.ora` by changing `SQLNET.AUTHENTICATION_SERVICES=(NTS) to SQLNET.AUTHENTICATION_SERVICES=(NONE).`

Setting Up Identity Management with NTFS Sources

If an NTFS source is used, Oracle recommends that Active Directory be used as identity management system for the Oracle SES instance. The Active Directory instance must be the same one that NTFS is using to authenticate users on the file system.

For the Oracle SES instance to read the files during crawling, add permission to each folder and file to make them accessible by the operating system user that runs the Oracle SES instance. (Adding permissions to a folder will automatically add the same permissions to all the files and sub-folders in the folder.)

See Also: "Activating an Identity Plug-in" on page 4-5 for information on activating the Active Directory identity plug-in

Creating an NTFS Source

Create an NTFS source on the **Home - Sources** page. Select NTFS from the Source Type list, and click **Create**. Enter the values for the following parameters:

Suppose you want to crawl `\\myserver\test1` and `\\myserver\test2` on an NTFS box. Specify the UNC PATH as follows: `\\myserver\test1` and `\\myserver\test2`. The domain user must have read privileges on the shared folders.

Setting Up Boundary Rules on NTFS Sources

Use boundary rules on the NTFS source to restrict the Oracle SES crawler to URLs that match the indicated rules. This is set on the **Home - Sources - Boundary Rules** page.

For simple rules, Oracle SES supports the *, ^, and $ special characters:

- SIMPLE_INC <simple boundary rule string>
- SIMPLE_EXC <simple boundary rule string>

This is a set of user-friendly, simplified regular expression rules. Specify an inclusion rule that a URL contain, start with, or end with a term. Use an asterisk (*) to represents a wildcard. Use a caret (^) to denote the beginning of a URL, and use a dollar sign ($) to denote the end of a URL. For example:

```
^https://*.oracle.com/
.jpg$
```

For regexp rules, Oracle SES supports all regexp patterns:

- REGEXP_INC <regular expression boundary rule string>
- REGEXP_EXC <regular expression boundary rule string>

This is a set of regular expression rules using the java.util.regex package.

For example:

```
^https://.*\.oracle(?:corp){0,1}\.com
```

For any of these parameters, you can specify up to 50 rules. Use a semi-colon to separate strings and specify multiple rules. For example:

```
/^https://.*\.oracle(?:corp){0,1}\.com;^https://*.oracle.com/;https://*.oracle.com
/*/
```

Setting Up Secure NTFS Sources for UNIX

This section includes information for Windows NT File System (NTFS) source on UNIX. NTFS sources for UNIX have additional setup steps not required on Windows.

An NTFS source collects the content, metadata attributes, and ACLs of files in NTFS. An NTFS source supports incremental crawl. After the initial crawl is performed, subsequent crawls only collect those documents that have changed since the last crawl. A document is re-crawled if the content, metadata or the ACL information of the document has changed. A file is also re-crawled if it is moved between folders. Files deleted from NTFS are removed from the index during incremental crawls.

Important Notes for NTFS Sources

- On the Windows server, the Super User must have permissions to read the NTFS file share
- The Super User must be the impersonate user in the IIS Server

Required Software

- Microsoft Internet Information Server (IIS)
- NET 2.0 Framework
- Microsoft Developer Support OLE File Property Reader (dsofile)

Required Tasks

NTFS sources on UNIX requires an NTFS Agent to be installed and configured on the Windows domain where the NTFS files are to be crawled. The NTFS Agent collects and sends content and meta data to the crawler plug-in on the Oracle SES machine in a crawl session. The communication protocol between Oracle SES and the NTFS Agent is HTTP or HTTPS.

The NTFS Agent needs to be installed on a Windows machine where IIS is present and the machine needs to be in the same Windows domain where the NTFS file share to be crawled resides.

Typically, a remote file share is crawled with the permission of a domain Administrator or a domain user with read privileges on the file share. The easiest way to configure this is to add the domain admin group to the 'administrators' group of the target machine.

The Oracle SES instance needs to connect to the same Active Directory instance that the MS NTFS domain connects to.

Install NTFS Agent on the Windows machine

1. If not already installed, download and install the Windows .Net 2.0 Framework.
2. If not already installed, download and install Microsoft Developer Support OLE File Property Reader.

3. Copy `dsofile.dll` to a Windows system folder on the machine where the IIS is installed. Register `dsofile.dll` file using `regsvr32.exe`. This machine will be where the NTFS Agent resides.

   See Also: `http://support.microsoft.com/?kbid=224351`

4. Configure NTFS Agent in IIS:

   a. Unzip `$ORACLE_HOME/search/lib/plugin/ntfsLinWin/NTFSWebService.zip` into a temporary directory

   b. Create a Virtual Directory in IIS and copy all the files unzipped from `NTFSWebService.zip` into the Virtual Directory, or copy the files into an existing Virtual Directory on IIS.

   c. For help in Creating Virtual Directories in IIS (IIS 6.0) see http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/5adfcce1-030d-45b8-997c-bdbfa08ea459.mspx?mfr=true

5. (Optional) Configure IIS Web site to use SSL

   See Also:

   - Configuring IIS Web site to use SSL: `http://www.petri.co.il/configure_ssl_on_your_website_with_iis.htm`
   - How to implement SSL in IIS: `http://support.microsoft.com/kb/299875`

6. Configure the NTFS Agent to connect to the NTFS store in IIS:

   a. Right-click your Web site (The IIS virtual directory with `NTFSWebService Folder/files`)

   b. Click the **Properties** tab.

   c. Click the **ASP.NET** button and Click **Edit Configurations**.

   d. ASP.NET Configuration/Application settings Parameters needs to be given

   Service UserName: User name to authenticate between Oracle SES and NTFS Agents. This user name is required in Oracle SES source configuration.

   Service Password: Password to authenticate between Oracle SES and NTFS Agents. This password is required in the Oracle SES source configuration.

   e. Configure ASPNET impersonation: Impersonation is performed when ASP.NET executes code in the context of an authenticated and authorized client. Using impersonation, ASP.NET applications can optionally execute the processing thread using the identity of the client on whose behalf they are operating.Configure IIS virtual Directory as follows:

   Right-click your IIS Web site (virtual directory), and then click **Properties**.

   Click the **ASP.NET** button and click **Edit Configurations**.

   Click the **Application** tab of ASP.NET Configuration Settings for Location Impersonation settings User Name: DOMAIN\<domain user>Password: password for <domain user>.

   NTFS Agent can be deployed in any IIS instance in the same Windows domain. Application user or super user (Impersonate User) must have read

permissions on the file share to be crawled. To enable read permissions do the following:

Right-click the file folder

Click **Properties**

Click security and then click **Advanced** tab.

Click effective permissions.

Enable read permissions for the user entered in the NTFS agent configuration.

Setting Up Identity Management with NTFS Sources

If an NTFS source is used, Oracle recommends that Active Directory be used as identity management system for the Oracle SES instance. The Active Directory instance must be the same one that NTFS is using to authenticate users on the file system.

For the Oracle SES instance to read the files during crawling, add permission to each folder and file to make them accessible by the operating system user that runs the Oracle SES instance. (Adding permissions to a folder will automatically add the same permissions to all the files and sub-folders in the folder.)

> **See Also:** "Activating an Identity Plug-in" on page 4-5 for information on activating the Active Directory identity plug-in

Creating an NTFS Source

Create an NTFS source on the **Home - Sources** page. Select NTFS from the Source Type list, and click **Create**. Enter the values for the following parameters:

- **UNC PATH**: UNC path for the NTFS system to crawl; for example, `\\MYSERVER\mysharedfolder`
- **EndPoint**: Target end point (HTTP or HTTPS); for example, `http(s)://NTFS Domain server (mail.doklet.com in this fig.)/virtual directory (NTFSWebService in the fig.)/NTFSWebService.asmx`
- **USER NAME**: User name to authenticate between Oracle SES and Microsoft Exchange: (configuration parameters similar to Exchange Agent in IIS)
- **PASSWORD**: Password to authenticate between Oracle SES and Microsoft Exchange: (configuration parameters similar to Exchange Agent in IIS)

Setting Up Boundary Rules on NTFS Sources

Use boundary rules on the NTFS source to restrict the Oracle SES crawler to URLs that match the indicated rules. This is set on the **Home - Sources - Boundary Rules** page.

For simple rules, Oracle SES supports the *, ^, and $ special characters:

- SIMPLE_INC <simple boundary rule string>
- SIMPLE_EXC <simple boundary rule string>

This is a set of user-friendly, simplified regular expression rules. Specify an inclusion rule that a URL contain, start with, or end with a term. Use an asterisk (*) to represents a wildcard. Use a caret (^) to denote the beginning of a URL, and use a dollar sign ($) to denote the end of a URL. For example:

```
^https://*.oracle.com/
.jpg$
```

For regexp rules, Oracle SES supports all regexp patterns:

- REGEXP_INC <regular expression boundary rule string>
- REGEXP_EXC <regular expression boundary rule string>

This is a set of regular expression rules using the java.util.regex package.

For example:

```
^https://.*\.oracle(?:corp){0,1}\.com
```

For any of these parameters, you can specify up to 50 rules. Use a semi-colon to separate strings and specify multiple rules. For example:

```
/^https://.*\.oracle(?:corp){0,1}\.com;^https://*.oracle.com/;https://*.oracle.com
/*/
```

Setting Up Secure Open Text Livelink Sources

Livelink data is stored in Workspaces, which in turn can contain folders, files, projects, and task lists. A Livelink Enterprise Server instance can have one or more Workspaces that can be crawled using the Livelink Enterprise Server plug-in by configuring the configuration parameter in Oracle SES. The Livelink Enterprise Server plug-in navigates through the Workspaces to crawl all the objects in Livelink Enterprise Server. It creates an index, stores the metadata, and accesses information in Oracle SES to provide search according to the end user permissions.

Important Notes for Open Text Livelink Sources

- The admin account should be used by the Livelink crawler plug-in for the container for crawling and indexing documents.
- The Livelink Enterprise Server version must be 9.2, 9.5.0, 9.5.5

Required Tasks

Because Open Text Livelink software is not included with Oracle SES, certain files must be copied manually into Oracle SES. Copy the `lapi.jar` file from LAPI installation folder into `ORACLE_HOME/search/lib/plugins/llcs`.

The Directory Services module of Livelink should be installed with Livelink (if users/groups are importing from LDAP server and you want to use the Active Directory identity plug-in).

To import users/groups of Active Directory in Livelink, follow these steps to import users/groups of Active Directory in Livelink Server.

Importing Users/Groups from LDAP to Livelink

1. Create an LDAP user that has permissions in Active Directory to administer users and groups. This user is used to synchronize the Active Directory with Livelink.
2. To extend the schema of Active Directory, install the Active Directory Schema snap-in as under:
   a. Select **Run** from Windows **Start** menu.
   b. Type `mmc /a` in the **Open** field and click **OK**.
   c. On the Console menu, choose **Add/Remove Snap-in** and click **Add**.

d. Under **Snap-in**, double-click **Active Directory Schema**. Click **Close**, then **OK**. Save the console (for example, as "Active Directory Schema.msc"). If the new snap-in does not appear under **Snap-in**, then you may have to re-install the Windows 2003 Administrative Tools and start again at step 2.

3. Open the file `ot-livelink-schema.conf` (it is in the directory `<livelink_home>/ module/directory_2_3_0`) in a text editor.
4. Open the **Active Directory Schema** console by clicking the Windows Start button, pointing to Programs - Administrative Tools and selecting (based on the sample name given) Active Directory Schema.msc.
5. Right-click **Active Directory Schema** and select **Operations Master**.
6. Right click the **Attributes** folder and select **Create Attribute**.
7. Create the attribute `llserverinfo` using the information from `ot-livelink-schema.conf` as under:

Table 5–6

Common Name	llserverinfo
LDAP Display Name	llserverinfo
Object ID	<Oracle Internet Directory> from `ot-livelink-schema.conf`
Syntax	Case Insensitive String
Multivalued	checked

8. Create the attribute `llquery` using the information from `ot-livelink-schema.conf` as under:

Table 5–7

Common Name	llquery
LDAP Display Name	llquery
Object ID	<OID>from ot-livelink-schema.conf
Syntax	Case Insensitive String
Multivalued	unchecked

9. Browse through the Directory Services Administration section of the Livelink Administration page for the enabling the following configuration:
   a. Enabling the Synchronization Features:

      Click the **Choose Directory Services** link.

      Select **LDAP Synchronization (Read-Only LDAP)** from the **Synchronization** list.

      For **Livelink CGI Hosts**, specify `127.0.0.1,<LIVELINK_SERVER_IP>`

      Click **Save Changes**.

   b. Configuring LDAP Read-Only Parameters:

Table 5–8

New User Password Policy	Hidden

Table 5–8 (Cont.)

User name Case Sensitivity	Preserve Case
Livelink Server Name	Machine name on which Livelink Server is running
LDAP Server	Machine name or IP Address on which LDAP Server is running
LDAP Server Port	389
Search Root	cn=Users,dc=otdomain,dc=com
LDAP User name	cn=<LDAP_User_Name>,cn=Users, dc=otdomain,dc=com
LDAP Password	<LDAP_User_Password>
Log-in Name	sAMAccountName or cn
First Name	givenname
Last Name	sn
Title	title
E-mail	mail
Contact	telephonenumber
Department Mapping	disable
Group Name	cn
Group Leader	managedBy
Group Member	Member
Group Member Query	llquery
Privileges	Select Log-in enabled, Public Access
Group Search Filter	objectclass=group
Synchronize Group	checked

Click **Save Changes**.

c. Click **Synchronize LDAP Read-only**.

Click **Synchronize**.

Known Limitations

If you update the attribute list, then you must update the crawler re-crawl policy to **Process All Documents** on the **Home - Schedules - Edit Schedules** page, and re-crawl the source.

Setting Up Identity Management for Open Text

The Livelink Enterprise Server identity plug-in authenticates native users of Livelink Enterprise Server. The identity plug-in communicates with the directory to authenticate a user's credentials, validate a user or group and return the associated canonical form, and return the groups associated with a given user.

Activate the identity plug-in on the **Global Settings - Identity Management Setup** page:

- For the Active Directory identity plug-in, activate the oracle.search.plugin.security.idm.IdentityPluginManagerADImpl plug-in.
- For the Livelink identity plug-in, activate the Livelink identity plug-in manager.

Creating an Open Text Livelink Source

Create an Open Text source on the **Home - Sources** page. Select Open Text from the Source Type list, and click **Create**. Enter values for the following parameters:

- **User name**: Name of a valid Livelink Enterprise Server user. The user must be an Administrator user or a user who has access to all folders and documents of the workspaces configured in the **Container name** parameter. The user should be able to retrieve content, metadata, and ACL from folders, documents and other custom sub classes of all workspaces configured in **Container name** parameter. This is a required parameter.
- **Password**: Password of the Livelink user. This is a required parameter.
- **Container name**: The names of the containers to be crawled by Oracle SES. You can crawl an entire Livelink Workspace or a specific folder. The format for is: <Workspace Name>/<Folder Name>/<Sub Folder Name>. Multiple comma-delimited container names can be entered. This is a required parameter. For example:
  - Container name: Workspace1: The entire Workspace1 will be crawled.
  - Container name: Workspace2/Folder21: Folder21 and its sub-folders within Workspace2 will be crawled.
- **Crawl folder attributes**: Indicate whether folder attributes need to be crawled, either true or false. This is an optional parameter. The default value is false. If any other value is provided, it is assumed to be false.
- **Crawl versions**: Indicates whether multiple versions of documents should be crawled, either true or false. This is an optional parameter and the default value is false. If any other value is provided, it is assumed to be false; in this case, only latest versions of a document will be crawled.
- **Attribute list**: The comma-delimited list of Livelink attributes along with their data types to be searchable. The format for attribute list is <Attribute Name>:<Attribute Type>, <Attribute Name:Attribute Type>. Valid values are String, Number, and Date.

Table 5–9 Open Text Data Types

Sr. No	Open Text Data Type	Oracle SES Data Type
1	Boolean	String
2	Integer	Number (Big Decimal)
3	String	String
4	Date	Date

While crawling a Workspace an attribute is indexed only if both name and type match with configured name and type; otherwise, it will be ignored. This is an optional parameter. For example: If the administrator wants to make the following Livelink attributes searchable:

- Attribute Name: Account Name Attribute Type: String
- Attribute Name: Account Id Attribute Type: Integer
- Attribute Name: Creation Date Attribute Type: Date

The value of **Attribute list** should be

Account Name: String, Account Id: Number, Creation Date:Date

The default searchable attributes for Livelink Enterprise Server will be Modified Date, Title, and Author.

Multiple attributes with same name are not allowed. For example Emp_ID:String, Emp_ID:Number

- **Server Name and Port Number for Livelink**: The machine name/IP address and the port number on which Livelink server is running. The format is <Server Name>:<Port Number>.
- **Authentication attribute**: The attribute used to set ACL. With Active Directory, the value is `USER_NAME`. With the Livelink identity plug-in, the value is `NATIVE`. This is a required parameter. This parameter is case-sensitive.
- **Crawl objects with public access**: This parameter indicates whether objects with public access should be crawled without any ACL. Valid values are true or false. If false, then all objects having this ACL will be ignored.
- **SSL Enabled for Livelink**: Specify if Livelink is running on SSL. If it is running on SSL, then this is true; otherwise, false.

Setting Up Secure Oracle Calendar Sources

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

Setting Up Identity Management for Oracle Calendar

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. On the **Global Settings - Identity Management Setup** page in the Oracle SES administration tool, select the **Oracle Internet Directory identity plug-in manager**, and click **Activate**.
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 with an LDAP client.)

   ```
   $ORACLE_HOME/bin/ldapmodify -h oidHost -p OIDPortNumber -D "cn=orcladmin" -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=ocsCsPlugin,cn=ifs,cn=Products,cn=OracleContext`. The entity will have the password `welcome1`.

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

Creating an Oracle Calendar Source

Create an Oracle Calendar source on the **Home - Sources** page. Select Oracle Calendar from the Source Type list, and click **Create**. Enter values for the following parameters:

Table 5–10 Calendar Source Parameters

Parameter	Value
Calendar server	http://*host name*:*port*
Application entity name	orclApplicationCommonName=ocsCsPlugin,cn=ifs,cn=Products,cn=OracleContext
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

Setting Up Secure Oracle Content Database Sources

Document 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.

Important Notes for 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.

Known Limitations

- 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 Known Limitations on page 5-28 and confirm that the number of crawler threads does not exceed the available concurrent connection settings for each user in Oracle Content Database.
2. 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.
3. 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 with an LDAP client.)

   ```
   $ORACLE_HOME/bin/ldapmodify -h oidHost -p OIDPortNumber -D "cn=oracle" -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 E, "LDIF Files" to view the `csPlugin.ldif` file

Creating an Oracle Content Database Source

Create an Oracle Content Database source on the **Home - Sources** page. Select Oracle Content Database from the Source Type list, and click **Create**. Enter values for the following parameters:

Table 5–11 Oracle Content Database Source Parameters

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

Table 5–12 Oracle Content Database Authorization Manager Plug-in Parameters

Parameter	Value
Oracle Content Database URL	http://*host name:port*/content
Oracle Content Database admin user	orcladmin
Entity name	`orclapplicationcommonname=ocscsplugin, cn=ifs,cn=products,cn=oraclecontext`
Entity password	welcome1
Use e-mail for authorization	false

Setting Up Secure Oracle E-Business Suite 11i Sources

An Oracle E-Business Suite 11i source crawler is based on crawling a view or query in a database. Each record in the view or query is considered a document.

Important Notes for Oracle E-Business Suite 11i Sources

The view or query to be crawled for this source should contain the following columns:

Table 5–13 Oracle E-Business Suite 11i Source Required Columns

Name	Type	Description
`URL`	varchar2	Display URL for the document
`SOLUTION`	varchar2/clob	Document content
`LASTMODIFIEDDATE`	date	Last modified date for crawls
`KEY`	varchar2	Key to the record
`LANG`	varchar2	Document language

The view or query can contain the following optional columns:

Table 5–14 Oracle E-Business Suite 11i Source Optional Columns

Name	Type	Description
PATH	varchar2	Path to the document. This is used in the browse feature.
ATTACHMENT_LINK	varchar2	HTTP link to the attachment for the document. This attachment will be indexed instead of the SOLUTION column.
ATTACHMENT	blob	Binary attachments for the document. This will be indexed instead of the SOLUTION column. This attachment will be indexed only if attachment link is not specified or the attachment pointed to by the link is not accessible.
CONTENTTYPE	varchar2	Content type of the text content (text/plain or text/HTML). This column can also be used to indicate the content type (if known) for the binary content.

Any other column in the view or query is considered an attribute of the document.

Setting Up Identity Management for Oracle E-Business Suite 11i

Activate the identity plug-in on the **Global Settings - Identity Management Setup** page. Select **Identity Plugin Manager for Oracle E-Business Suite 11i** and click **Activate**. Enter the values for the following parameters:

- **User Validation Database Connection String**: JDBC connection string for the database, used for validating a user.
- **User ID**: User ID to login to the user validation database.
- **Password**: Password to login to the user validation database.
- **User Authentication Query**: SQL query to authenticate a user. The query should return a single record with a single column with a string value of 'Y' or 'N' based on successful or unsuccessful authentication, respectively. The placeholder for user name and password should be specified as '?'. The default query (which can be changed if needed) is:

```
SELECT fnd_web_sec.Validate_login(upper(?),?)
FROM dual
```

- **User Validation Query**: SQL query to validate a given user. The query should return 1 if the user is valid. Else, no rows should be returned. The placeholder for the user name should be specified as '?'. The default query (which can be changed if needed) is:

```
SELECT 1
FROM fnd_user
WHERE user_name = upper(?)
```

Click **Finish**.

Creating an Oracle E-Business Suite 11i Source

Create an Oracle E-Business Suite 11i source on the **Home - Sources** page. Select **Oracle E-Business Suite 11i** from the Source Type list, and click **Create**. Enter values for the following parameters:

- **Database Connection String**: JDBC connection string for the E-Business Suite database from which the content will be crawled.
- **User ID**: User ID to login to the E-Business Suite database. This user ID should have access to the schema owning the view specified in the **View** parameter.
- **Password**: Password to login to the E-Business Suite database.
- **View**: Table or view containing the required set of columns
- **Document Count**: Maximum number of documents to be crawled and indexed. Enter `-1` if all documents should be crawled before indexing.
- **Query**: Query projecting the required set of columns. This query should be used if the view defined in the **View** parameter is not available. Only one of these - **View** or **Query** – should be specified.
- **URL Prefix**: String to prefix the content of URL column to form a display URL for the document
- **Cache File**: Local file to which the contents can be temporarily cached while crawling.
- **Path Separator**: Path separator character in the document path string
- **Parse Attributes**: Enter true if the values of the attributes should be extracted from the document content specified in `SOLUTION` column. Otherwise, enter false.
- **Grant Security Attributes**: Space-delimited list of grant security attributes
- **Deny Security Attributes**: Space-delimited list of deny security attributes

Click **Next**.

Click **Get Parameters** to obtain a list of parameters for the authorization manager plug-in.

Enter the values for the authorization manager plug-in parameters:

- **Authorization Database Connection String**: JDBC connection string for the authorization database. The values of the security attributes to which a given user is authorized will be retrieved from this database.
- **User ID**: User ID to login to the authorization database
- **Password**: Password to login to the authorization database
- **Authorization Query**: SQL query to retrieve the values of security attributes to which a given user is authorized. The `SELECT` clause of this query should have all the security attributes specified in the **Grant Security Attributes** and **Deny Security Attributes** parameters with identical names. This query can be of two types:
  - The query can return a single record for a given user. The value in each security attribute column should be a space-delimited list of values to which the user is authorized.
  - The query can return multiple records for a given user. The value in each security attribute column of every row of the result set of this query will be interpreted as a single value.

  The placeholder for the user name in the query should be specified as '?'.
- **Single Record Query**: Enter true if the authorization query returns a single record. Enter false if the query can return multiple records.

Click **Create**.

Setting up Secure Siebel 8 Sources

For Siebel sources, searching is based on Siebel data available as RSS feeds. This section provides the instructions to create a secure Siebel 8 source.

Setting Up Identity Management for Siebel 8

Activate the identity plug-in on the **Global Settings - Identity Management Setup** page. Select **Identity Plugin Manager for Siebel 8** and click **Activate**.

1. Enter values for the following parameters:
   - **Siebel 8 authentication Web service endpoint**: HTTP endpoint of the Siebel Web service that provides the authentication service
   - **Siebel 8 validation Web service endpoint**: HTTP endpoint of the Siebel Web service that provides the user validation service
   - **User ID**: Admin user ID for accessing the user validation service
   - **Password**: Admin password for accessing the user validation service
2. Click **Finish**.

Creating a Siebel 8 Source

Create a Siebel 8 source on the **Home - Sources** page. Select **Siebel 8** from the Source Type list, and click **Create**.

1. Enter the values for the following parameters:
   - **Configuration URL**: File URL of the XML configuration file providing details about the source, such as the data feed type, location, security attributes, and so on.

     Obtain this file from Siebel administrator and save it on the machine on which Oracle SES is installed. Enter the configuration URL as `file://localhost/<Absolute path of the configuration file>`. For example: `file://localhost/private/oracle/config.xml/`.
   - **User ID**: User ID to login to the FTP server, if the data feeds are to be accessed over FTP. The access details of the data feed are specified in the configuration file. This can be obtained from Siebel administrator.
   - **Password**: Password to login to the FTP server. This can be obtained from Siebel administrator.
   - **Scratch Directory**: A directory, in the machine where Oracle SES is installed to temporarily write the status logs.
   - **Maximum number of connection attempts**: Maximum number of attempts to connect to the target server to access the data feed.
2. Click **Next**.
3. Enter the values for the authorization manager plug-in parameters:
   - **Siebel 8 authorization Web service endpoint**: Webs service endpoint of the Siebel Web service that provides the authorization service
   - **User ID**: Admin user ID for accessing the authorization service
   - **Password**: Admin password for accessing the authorization service

4. Click **Create**.

Setting Up Secure Microsoft Exchange Sources

Oracle SES can crawl through the e-mail and calendar items, related metadata, attributes, ACLs and attachments in Exchange and provide secure search. It also provides attribute search and browse functionality, which allows search to be done against a specific subfolder in the hierarchy.

The Microsoft Exchange plug-in supports incremental crawling; that is, it crawls and indexes only those documents that have changed after the last crawl was scheduled. A document is re-crawled if either the content or metadata or the direct security access (permissions) information of the document has changed. A document is also re-crawled if it is moved within Microsoft Exchange. Documents deleted from Exchange are removed from the index during incremental crawls.

A Microsoft Exchange source covers the following objects in Exchange:

- E-mail
- E-mail attachments
- Calendar events

Important Notes for Microsoft Exchange Sources

On the Exchange server, the super user needs to grant himself the `Send as` and `Receive as` privileges. You can enable privileges globally for all users in the system. No user-specific privilege grants are required.

See Also:

- *Microsoft Exchange 2003 Technical Reference Guide* and information about permissions in Microsoft Exhange: `http://www.microsoft.com/technet/prodtechnol/exchange/default.mspx`
- *Oracle Secure Enterprise Search Release Notes* for supported platforms

Required Software

- Microsoft Internet Information Server (IIS)
- NET 2.0 Framework

Required Tasks

Proper permissions on the Exchange server need to be granted to the Exchange administrator. The Exchange server is crawled with the permission of a super user with the `Send as` and `Receive as` privileges. The easiest way to configure this is to use an administrator as super user or create a super user with the administrator privilege and the `Send as` and `Receive as` privileges targeting Exchange inbox store and public folders.

The Microsoft Exchange source requires an *Exchange Agent* to be installed and configured on the Windows domain where the Exchange server is to be crawled. The Exchange Agent collects and sends content and metadata to the crawler plug-in on the Oracle SES machine in a crawl session. The communication protocol between Oracle SES and the Exchange Agent is HTTP or HTTPS.

The Exchange Agent must be installed on a Windows machine where IIS is present, and the machine needs to be in the same Windows domain where the Exchange server to be crawled resides.

Install the Exchange Agent on the Exchange server:

1. Unzip `$ORACLE_HOME/search/lib/plugin/msexchange/ExchangeWebService.zip` into a temporary directory.
2. Create a virtual directory in IIS (IIS 6.0) and copy all the files unzipped from `ExchangeWebService.zip` into the virtual directory, or copy the files into an existing virtual directory on IIS.

   See Also:

   http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/5adfcce1-030d-45b8-997c-bdbfa08ea459.mspx?mfr=true·

3. (Optional) Configure IIS Web site to use SSL:

   See Also:

   - Configuring IIS Web site to use SSL: http://www.petri.co.il/configure_ssl_on_your_website_with_iis.htm
   - How to implement SSL in IIS: http://support.microsoft.com/kb/299875

4. Configure the Exchange Agent to connect to native Exchange Server store:
   a. Right-click your Web site (the IIS virtual directory with Exchange Agent files).
   b. Click the **Properties** tab.
   c. Click the **ASP.NET** button, and click **Edit Configurations**.
   d. Application settings parameters must be entered:

      Service UserName: User name to authenticate between Oracle SES and Exchange Agent. This user name is required in Oracle SES source configuration.

      Service Password: Password to authenticate between Oracle SES and Exchange Agent. This password is required in the Oracle SES source configuration.

5. Enter impersonation settings. Impersonation is when ASP.NET executes code in the context of an authenticated and authorized client. Using impersonation, ASP.NET applications can optionally execute the processing thread using the identity of the client on whose behalf they are operating. Configure IIS virtual Directory as follows:
   a. Right-click your IIS Web site (virtual directory), and then click **Properties**.
   b. Click the ASP.NET button, and click **Edit Configurations**.
   c. Click the **Application** tab of ASP.NET Configuration Settings for Location Impersonation settings:

      User Name: DOMAIN\SuperUser

Password: Password for SuperUser

The Exchange Agent can be deployed in any IIS in the same Windows domain.

Setting Up Identity Management for Microsoft Exchange

If a Microsoft Exchange source is used, Oracle recommends that Active Directory be used as identity management system for the Oracle SES instance. The Active Directory instance must be the same one that Microsoft Exchange is using to authenticate users on the file system.

For the Oracle SES instance to read the files during crawling, add permission to each folder and file to make them accessible by the operating system user that runs the Oracle SES instance. (Adding permissions to a folder will automatically add the same permissions to all the files and sub-folders in the folder.)

> **See Also:** "Activating an Identity Plug-in" on page 4-5 for information on activating the Active Directory identity plug-in

Creating a Microsoft Exchange Source

Create a Microsoft Exchange source on the **Home - Sources** page. Select **Microsoft Exchange** from the Source Type list, and click **Create**.

Enter values for the following parameters:

- **USER NAME**: User name to authenticate between Oracle SES and Exchange (configuration parameters consistent with that for Exchange Agent in IIS).
- **PASSWORD**: password to authenticate between Oracle SES and Exchange (configuration parameters consistent with that for Exchange Agent in IIS).
- **ENDPOINT**: Target end point (HTTP or HTTPS); for example, http(s)://exchange server (mail.doklet.com in the example)/virtual directory (Web site in the example)/ExchangehttpsService.asmx.

Setting Up Boundary Rules on Microsoft Exchange Sources

Use boundary rules on the Microsoft Exchange source to restrict the Oracle SES crawler to URLs that match the indicated rules. This is set on the **Home - Sources - Boundary Rules** page.

For simple rules, Oracle SES supports the *, ^, and $ special characters:

- SIMPLE INCLUDE <simple boundary rule string>
- SIMPLE EXCLUDE <simple boundary rule string>

This is a set of user-friendly, simplified regular expression rules. Specify an inclusion rule that a URL contain, start with, or end with a term. Use an asterisk (*) to represents a wildcard. Use a caret (^) to denote the beginning of a URL, and use a dollar sign ($) to denote the end of a URL. For example:

```
^https://*.oracle.com/
.jpg$
```

For regexp rules, Oracle SES supports all regexp patterns:

- Regular Expression INCLUDE <regular expression boundary rule string>
- Regular Expression: EXCLUDE <regular expression boundary rule string>

This is a set of regular expression rules using the `java.util.regex` package.

For example:

```
^https://.*\.oracle(?:corp){0,1}\.com
```

For any of these parameters, you can specify up to 50 rules. Use a semi-colon to separate strings and specify multiple rules. For example:

```
/^https://.*\.oracle(?:corp){0,1}\.com;^https://*.oracle.com/;https://*.oracle.com
/*/
```

Setting Up Secure Federated Sources

See Also: "Tips for Using Federated Sources" on page 6-3

Secure federated search enables searching secure content across distributed Oracle SES instances. An end user is authenticated to the Oracle SES federation broker. Along with querying the secure content in its own index, the federation broker federates the query to each federation endpoint 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 Oracle SES instances. This section explains how Oracle SES performs secure federation.

See Also:

- "Configuring Secure Search with OracleAS Single Sign-On" on page 4-11
- Appendix A, "10.1.6 to 10.1.8 Upgrade"

Federation Trusted Entities

When performing a secure search on a federation endpoint, the federation broker must pass the identity of the logged in user to the federation endpoint. If the endpoint instance trusts the broker instance, then the broker 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.

Use **Authentication Attribute** to specify the format of the user credential that the Oracle SES federation endpoint expects for this particular trusted entity in proxy authentication. The identity plug-in registered on the federation endpoint should be able to map this user identity to the default authentication format used on the federation endpoint. This is useful when a federation broker cannot send user identity in the default authentication format used on the federation endpoint for proxy authentication, but the identity plug-in registered on the federation endpoint can map the value from the attribute in which it receives the user identity during proxy authentication to the default authentication format used on the federation endpoint.

To use a proxy entity, use the Web services API `proxyLogin()` user name and password for the entity name and entity password. The identity plug-in can validate the password instead of storing it. When a request is sent for `proxyLogin()`, Oracle SES calls the identity plug-in (which returns the call) to authenticate the entity. The `proxyLogin()` must supply one of the valid trusted entities registered in the federation trusted entities.

To perform secure federated search, both the broker and the endpoint 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 broker instance:

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

  The identity plug-in registered on the broker 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 broker instance is used to search on the endpoint 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 broker or the endpoint instance might or might not be using single sign-on (SSO). However, the Web service URL of the endpoint should not be behind SSO.
- Oracle strongly recommends that you SSL-protect the channel between Oracle HTTP Server and Oracle SES for secure content. The endpoint instance should be SSL-enabled, or you should be able to access the Web service using HTTPS.

See Also: "Tips for Using Federated Sources" on page 6-3

6

Oracle Secure Enterprise Search Advanced Information

This chapter contains the following topics:

Troubleshooting Sources

This section 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`, `BFILE`, `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 topics:

- 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/`.
- 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 6-3

Tips for Using Federated Sources

- 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.

- If you entered proxy settings on the **Global Settings - Proxy Settings** page, then make sure to add the Web Services URL for the federated source as a proxy exception.
- If the federation endpoint instance is set to secure mode 3 (require login to search secure and public content), then all documents (ACL stamped or not) are secure. For secure federated search, create a trusted entity in the federation endpoint instance, then edit the federated source with the trusted entity user name and password.

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 federation endpoint to be displayed on the Oracle SES federation broker instance.
- 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 a sequence number is added to the folder name belonging to the federated source to distinguish the two hierarchies on the Browse page.
- For federated infosource browse, a federated source should be put under an explicitly created source group.
- On the Oracle SES federation broker, there is no direct access to documents on the federation endpoint 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.

See Also:

- "Setting Up Secure Federated Sources" on page 5-37 if the federated source will be searching private content
- Appendix A, "10.1.6 to 10.1.8 Upgrade"

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:

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:

- Add Suggested Links or Suggested Content
- Optimize the Index
- Increase the Indexing Batch Size
- Increase the Index Memory Size
- Check the Search Statistics
- Increase the JVM Heap Size
- Increase the Oracle Undo Space

Add Suggested Links or Suggested Content

Suggested links let you direct users to a particular Web site for a given search string. For example, when users search for "Oracle Secure Enterprise Search documentation" or "Enterprise Search documentation" or "Search documentation", you could suggest

http://www.oracle.com/technology. Suggested links appear at the top of the search result list. This feature is especially useful to provide links to important Web pages that are not crawled by Oracle Secure Enterprise Search. Set suggested links on the **Search - Suggested Links** page in the administration tool.

Suggested content lets you display real-time data content in the result list of the default query application. Oracle SES retrieves data from content providers and applies a style sheet to the data to generate an HTML fragment. The HTML fragment is displayed in the result list and is available through the Web Services API. For example, when an end user searches for contact information on a coworker, Oracle SES can fetch the content from the suggested content provider and return the contact information (e-mail address, phone number, and so on) for that person in the result list. Suggested content results appear under any suggested links and above the query results.

Configure suggested content on the **Search - Suggested Content** page in the administration tool. Enter the maximum number of suggested content results (up to 20) to be included in the Oracle SES result list. The results are rendered on a first-come, first-served basis.

Regular expressions (as supported in the Java regular expression API `java.util.regex`) are used to define query patterns for suggested content providers. The regular expression-based pattern matching is case-sensitive. For example, a provider with the pattern `dir\s(\S+)` is triggered on the query `dir james` but not on the query `Dir James`. To trigger on the query `Dir James`, the pattern could be defined either as `[Dd][Ii][Rr]\s+(\S+)` or as `(?i)dir\s+(\S+)`. A provider with a blank query pattern is triggered on all queries.

The URL you enter for the suggested content provider can contain the following variables: $ora:q, $ora:lang, $ora:q1, ... $ora:qn and $ora:username.

- $ora:q is the end user full query.
- $ora:lang is the two-letter code for the browser language
- $ora:qn is the nth regular expression match group from the end user query. n starts from 1. If no nth group is matched, then the empty string replaces the variable.
- $ora:username is the end user name.

Enter an XSLT style sheet to defines rules (for example, the size and style) for transforming XML content from a provider into an HTML fragment. This HTML fragment is displayed in the result list or returned over the Web Services API. If you do not enter an XSLT style sheet, then Oracle SES assumes that the suggested content provider returns HTML. If you do not enter an XSLT style sheet and the provider returns XML, then the result list displays the plain XML.

> **Note:** It is the administrator's responsibility to ensure that suggested content providers return valid and safe content. Corrupted or incomplete content returned by an suggested content provider can affect the formatting of the default query application results page.

There are three security options for how Oracle SES passes the end user's authentication information to the suggested content provider:

- None: With this method (the default), no security policy is used.
- Cookie: With this method, the end user first must be authenticated by the suggested content provider. A cookie is set for the user to maintain a session. Oracle SES must know the cookie used by the provider for authentication, and it is

made available during registration of the suggested content provider. When the user enters a query, Oracle SES grabs the cookies from the user's request header and passes them to the provider. The cookie scope must be set to the common domain of the provider site and the Oracle SES site by the provider.

For example, suppose the provider site is http://provider.company.com and the Oracle SES site is http://ses.company.com. After the end user logs in to the provider site, the site could set the value of the security cookie loginCookie with domain scope .company.com. When the end user searches in Oracle SES, Oracle SES gets the loginCookie value from the end user browser and forwards it to the provider site to get the suggested content (without login to the provider site again). However, if the provider site is accessed as http://provider or if the Oracle SES site is accessed as http://SES, then no domain cookie is available for sharing between the two sites and this security mechanism does not work.

You can decide what happens when suggested content is available but the user is not logged in to the suggested content provider or the cookie for the provider is not available. For **Unauthenticated User Action**, if you select **Ignore content**, then content from that provider will not be displayed in the result list. If you select **Display login message**, then Oracle SES returns a message that there is content available from this provider but the user is not logged in. The message also provides a link to log in to that provider. Enter the link for the suggested content provider login in the **Login URL** field.

- Service-to-Service: With this method, a one-way trusted relationship is established between Oracle SES and the suggested content provider. Any user already logged in to Oracle SES does not need to be authenticated by the provider again. The provider only authenticates the Oracle SES application and trusts the Oracle SES application to act as the end user. The end user identity is sent from Oracle SES to the provider site in the HTTP header `ORA_S2S_PROXY_USER`. The trusted entity could be a proxy user configured in the identity management system used by the provider, or it could be a name-value pair.

Example Configuring Google OneBox for Suggested Content

Existing OneBox providers can be configured for use as Oracle SES Suggested Content providers. For example, for a Google OneBox provider, the provider URL might be `http://host.company.com/apps/directory.jsp` and the trigger might be `dir\s(\S+)`. When the user query is **dir james**, the provider receives the request with a query string similar to the following: `apiMaj=10&apiMin=1&oneboxName=app&query=james`.

With a Suggested Content provider, set the URL template as `http://host.company.com/apps/directory.jsp?apiMaj=10&apiMin=1&oneboxName=app&query=$ora:q1`. The provider pattern is the same: `dir\s(\S+)`. The XSLT used for Google OneBox can be re-used with a minor change. Look for the line:

```
<xsl:template name="apps">
```

and change that line in your template to

```
<xsl:template match="/OneBoxResults">
```

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 Indexing Batch Size

The data in the cache directory continues to accumulate until it reaches this limit. When the limit is reached, the data is indexed. 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.

The default indexing batch size is 250M. Increasing the size up to the index memory size (275M by default) can reduce index fragmentation. However, increasing the size more than the index memory size will not reduce fragmentation. You can change the index memory size manually.

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

Increase the Index Memory Size

A large index memory setting (even hundreds of megabytes) improves the speed of indexing and reduces the fragmentation of the final indexes. However, there will be a point where it is set so high that memory paging occurs and impacts indexing speed.

Follow these steps to increase the index memory size:

1. Launch SQL*Plus and connect as the `eqsys` user.
2. Run the following SQL statement to see the current indexing memory size:

```
SQL> SELECT par_value FROM ctx_parameters
2  WHERE par_name = 'DEFAULT_INDEX_MEMORY';

PAR_VALUE
-----------
288358400
```

   This is the default value for indexing memory size. The unit is bytes. (288358400 bytes = 275M bytes)

3. To change the default indexing memory size to 500M (524288000bytes), run the following procedure:

```
SQL> begin
2  ctxsys.ctx_adm.set_parameter('DEFAULT_INDEX_MEMORY','524288000');
3  end;
4  /

PL/SQL procedure successfully completed.

SQL> SELECT par_value FROM ctx_parameters
2  WHERE par_name = 'DEFAULT_INDEX_MEMORY';

PAR_VALUE
-----------
524288000
```

4. You can specify up to 2G for `DEFAULT_INDEX_MEMORY`. To allocate more than 1G, you also must change `MAX_INDEX_MEMORY`. `DEFAULT_INDEX_MEMORY`

cannot exceed MAX_INDEX_MEMORY, and the default value for MAX_INDEX_MEMROY is 1G. The maximum size for MAX_INDEX_MEMORY is 2,147,483,647 bytes.

```
SQL> begin
2  ctxsys.ctx_adm.set_parameter('MAX_INDEX_MEMORY','2147483647');
3  end;
4  /

PL/SQL procedure successfully completed.

SQL> begin
2  ctxsys.ctx_adm.set_parameter('DEFAULT_INDEX_MEMORY','2147483647');
3  end;
4  /

PL/SQL procedure successfully completed.
```

You can change the memory size any time. The next synchronized index uses this specified memory size.

Note: The indexing batch size determines when the synchronized index is called. Even if DEFAULT_INDEX_MEMORY is large enough, Oracle SES does not use it if the indexing batch size is small. For example, if the indexing batch size is 10M, then the synchronized index uses memory up to 10M, even if you specify 1G for it.

Tip: "Increase the Indexing Batch Size" on page 6-11

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.

Relevancy boosting, like end user searching, is case-insensitve. For example, a document with a boosted score for "Oracle" is boosted when you enter "oracle".

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

7

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 APIs
- Oracle Secure Enterprise Search Java 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 APIs

The Web Services APIs are used to integrate Oracle SES search capabilities into your search application. Oracle SES provides Java proxy libraries. You either can use the Java libraries or create proxies, based on the published Web Services Description Language (WSDL) files, to access Oracle SES Web Services.

The Query Web Service API lets you perform search queries; for example, search for "oracle benefits" and return all the documents.

The Admin Web Service API lets you perform a subset of administrative actions, such as starting and stopping a crawler schedule or getting the index fragmentation level.

> **See Also:** The "Web Services Interface" section in the Oracle SES administration tutorial:
>
> `http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm`

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 APIs

Oracle Secure Enterprise Search Web Services APIs let you write your own application to search and administer Oracle SES over the network. The APIs provide 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 topics:

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.

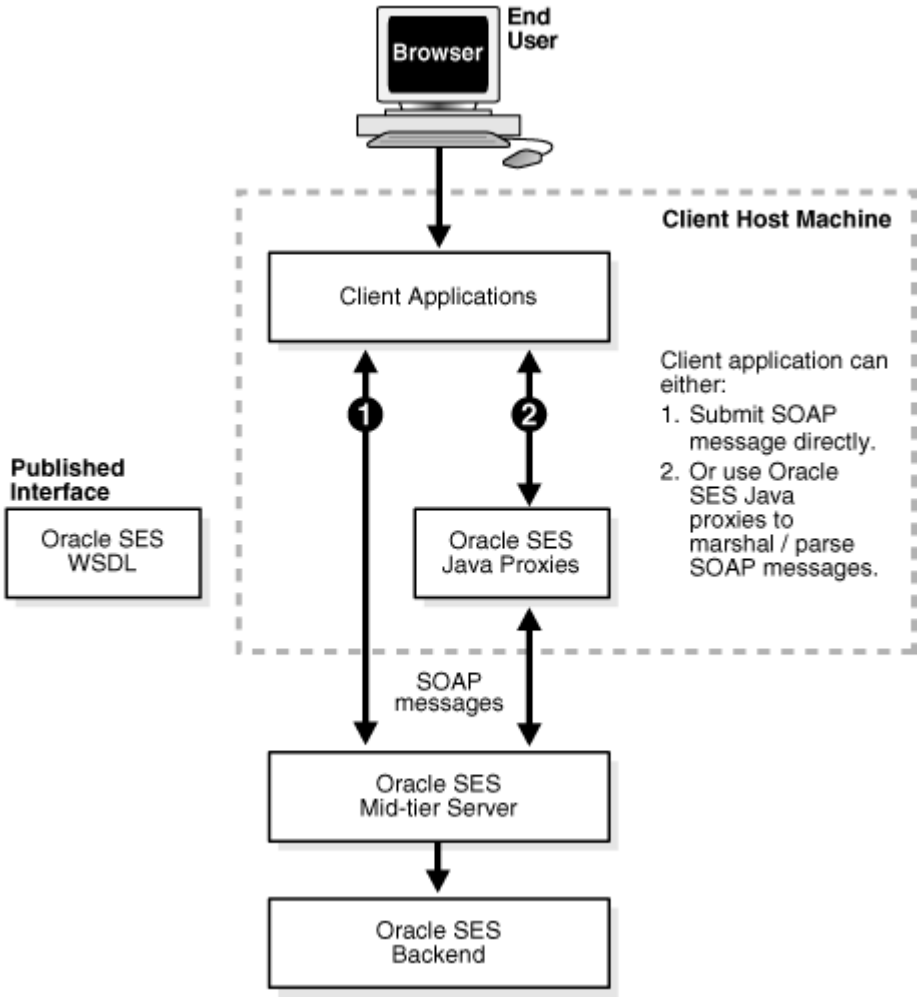
Oracle Secure Enterprise Search Web Services Architecture

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. Get suggested content from external providers for the given query.
- 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 Query Web Service Operations" on page 7-9

Oracle Secure Enterprise Search Web Services Common Data Types

This section contains the following topics:

Base Data Types

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

Table 7–1 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 7–2 XML-to-Java Type Mappings

XML Schema	Oracle JDeveloper
xsd:Boolean	java.lang.Boolean
xsd:date	java.util.Date

Table 7–2 (Cont.) XML-to-Java Type Mappings

XML Schema	Oracle JDeveloper
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 estimated 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

SCElement Suggested content from a provider. It has following elements:

- `name`: name of the suggested content provider
- `content`: suggested content from the provider. The content is a byte array of the XML or HTML content

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 `attributeType` is string, then it should be either equals or contains.
  - If `attributeType` is number or date, then it should be either greaterthan, greaterthanequals, lessthan, lessthanequals, 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 Secure Enterprise Search', 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 'ailed
- `message`: Status message. Value is null, or an error message if the status is 'ailed

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

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`
- `SCElementArray`: Array of `SCElement`
- `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 D, "WSDL Specifications"

Oracle Secure Enterprise Search Query Web Service Operations

This section contains the following topics:

- 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
- proxyLoginRequest Message
- proxyLoginResponse 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>
```

> **Note:** User name is *not* case-sensitive.

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 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 the 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>
```

proxyLoginRequest Message This message logs in the end user to Oracle SES using proxy authentication. It consists of following parameters:

- `username`: User name of the proxy user
- `password`: Password of the proxy user
- `searchUser`: User name of the end user

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

The proxy user must be one of the federation trusted entities created on the Oracle SES instance.

See Also: "Federation Trusted Entities" on page 5-37

proxyLoginResponse Message This message contains the return status for the `proxyLoginRequest` message.

```
<message name="proxyLoginResponse">
  <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
- getSuggestedContent Message
- getSuggestedContentResponse 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 Query Web Service Query Syntax" on page 7-20 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 the query. This is equivalent to locale. The default is English ("en") if not set explicitly. This is used for relevancy boosting.
- `docLang`: Set the language of the documents to limit the search. If the value is not set explicitly, then search is performed against documents of all the languages.
- `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 (or set one int value '0'), so 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 Query Web Service Query Syntax" on page 7-20 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 of the query. This is equivalent to locale. The default is English ("en") if not set explicitly. This is used for relevancy boosting.
- `docLang`: Set the language of the documents to limit the search. If the value is not set explicitly, then search is performed against documents of all the 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 (or set one int value '0'), so 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 Query Web Service Query Syntax" on page 7-20 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>
```

getSuggestedContent Message This message returns the suggested content for the given query. It consists of the following parameters:

- `query`: Query string
- `returnType`: Format in which the content is to be returned, either "html" or "xml". If no style sheet is configured for a given provider, then the return type is the return type of the content returned by the provider, regardless of whether "html" or "xml" is specified.

```
<message name="getSuggestedContent">
  <part name="query"                  type="xsd:string"/>
  <part name="returnType"             type="xsd:string"/>
</message>
```

getSuggestedContentResponse Message This message returns the suggested content for the query.

```
<message name="getSuggestedContentResponse">
  <part name="return"                 type="typens:SCElementArray"/>
</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 Query Web Service 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 in the search string: oracle, secure, enterprise, and search. If the search string is oracle "secure enterprise search", then there are two search terms in 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, htm, html, xml, ps, pdf, txt, rtf, ppt, and xls. doc, html, pdf, txt, rtf, ppt, 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 Query Web Service 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 Query Web Service 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://<host>:<port>/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 Query 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.query.webservice.client.Attribute`
- `oracle.search.query.webservice.client.AttributeLOVElement`
- `oracle.search.query.webservice.client.CustomAttribute`
- `oracle.search.query.webservice.client.DataGroup`
- `oracle.search.query.webservice.client.Filter`
- `oracle.search.query.webservice.client.Language`
- `oracle.search.query.webservice.client.Node`
- `oracle.search.query.webservice.client.OracleSearchResult`
- `oracle.search.query.webservice.client.OracleSearchService`
- `oracle.search.query.webservice.client.ResultElement`
- `oracle.search.query.webservice.client.SessionContextElement`
- `oracle.search.query.webservice.client.Status`
- `oracle.search.query.webservice.client.SuggestedLink`
- `oracle.search.query.webservice.client.SCElement`

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.

- `$ORACLE_HOME/search/lib/search_client.jar` (The proxy Java library)
- `$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 Query Web Service 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`

Oracle Secure Enterprise Search Admin Web Service Endpoint Location

The Admin Web service is located at the following address for an Oracle SES installation: `http://<host>:<port>/search/ws/admin/SearchAdmin`.

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

- Oracle SES Admin WSDL description
- List of Web Service messages and operations
- Client-side JavaScript stub

Client-Side Admin Java Proxy Library

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

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

- `oracle.search.admin.ws.client.Schedule`
- `oracle.search.admin.ws.client.ScheduleStatus`
- `oracle.search.admin.ws.client.SearchAdminClient`

To compile and run your client application using the Oracle SES client-side Java proxy stub, include the following files in the Java CLASSPATH:

- `$ORACLE_HOME/search/lib/search_admin_wsclient.jar`
- `wsclient_extended.jar`

The `wsclient_extended.jar` file is available as a separate download from the Oracle Technology network: `http://download.oracle.com/otn/java/oc4j/10131/wsclient_extended.zip`

See Also:

- *Oracle Secure Enterprise Search Java API Reference*
- "Setting the Classpath for a Web Service Proxy" in the Oracle Application Server Web Services Developer's Guide, 10g Release 3 (10.1.3.1.0)

Oracle Secure Enterprise Search Admin Web Service SOAP Fault Error Codes

If an error occurs as a result of an Admin Web Service request, a SOAP fault is returned. When using the provided Java proxy client, a `javax.xml.rpc.soap.SOAPFaultException` is thrown. To access the machine parseable error code, call the `SOAPFaultException.getFaultCode()` method.

The following table lists the Admin Web Service error codes:

Table 7–3 Admin Web Service Error Codes

Error Code	Description	SOAP Fault Code Prefix
`Authentication`	The provided security credentials are not valid	Client
`InternalError`	An internal error occurred. Please try again	Server
`InvalidSchedule`	The specified schedule is invalid for the operation performed.	Client
`InvalidScheduleName`	The specified schedule name does not exist.	Client

Oracle Secure Enterprise Search Java SDK

The Oracle Secure Enterprise Search Java 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.

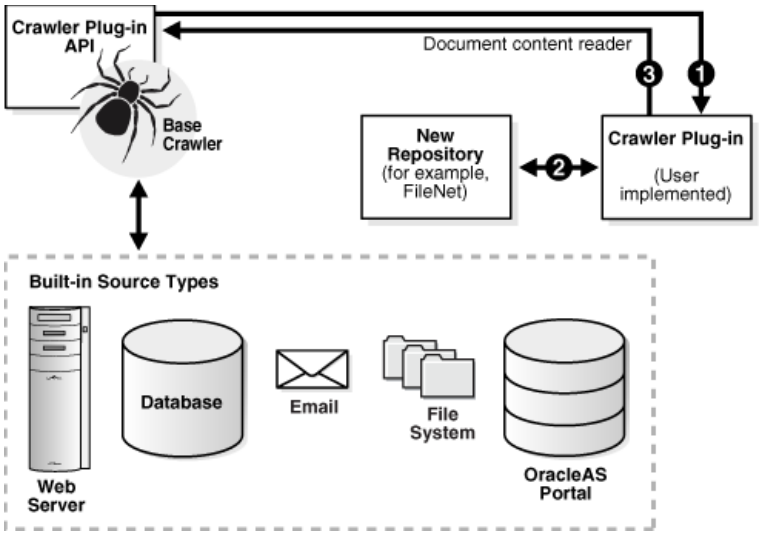
This section includes the following topics:

- 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 in 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 must be put under the `$ORACLE_HOME/lib` directory or the `$ORACLE_HOME/search/lib/plugins` directory. The Java library path is set explicitly by the crawler to those locations.

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 7–4 Crawler Plug-in Interfaces and Classes

Interface/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 interface 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.
`DataSourceService`	This interface 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 interface 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 interface 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.
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/plugins/` directory. For example:

   ```
   $ORACLE_HOME/jdk/bin/jar cv0f $ORACLE_HOME/search/lib/plugins/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.sdk.crawler` package)

Query-time Authorization API

Query-time authorization allows an Oracle 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. This result 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 Query Web Services interface. For either type of request, the key used to access the authenticated user name is the string value `AUTH_USER`.

> **Note:** User name is *not* case-sensitive.

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 )
  {
    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" );
  }

  if( user == null )
    user = "unknown";

  return user;
```

```
}
```

See Also: "Authentication Methods" on page 4-7

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 7–5 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

A

10.1.6 to 10.1.8 Upgrade

This appendix contains topics relating to an upgraded Oracle SES instance. This contains the following topics:

- Oracle Calendar Sources from 10.1.6
- Secure Federated Search Between Releases 10.1.8 and 10.1.6

See Also:

- *Oracle Secure Enterprise Search Installation and Upgrade Guide* for your platform for information about upgrading
- "Upgrade Issues" in the *Oracle Secure Enterprise Search Release Notes*

Oracle Calendar Sources from 10.1.6

Oracle Calendar sources created in Oracle SES 10.1.6 may not work after upgrade. 10.1.8 uses a newer version of OC4J, and the `soap.jar` file included in OC4J is in a different location.

- 10.1.6 `soap.jar` location: `$ORACLE_HOME/oc4j/soap/lib/soap.jar`
- 10.1.8 `soap.jar` location: `$ORACLE_HOME/oc4j/webservices/lib/soap.jar`

Create new Oracle Calendar sources in 10.1.8. Otherwise, to use the Oracle Calendar sources created in 10.1.6, create the directory structure identical to the 10.1.6 location (`*$ORACLE_HOME/oc4j/soap/lib/ *`) and put a copy of `soap.jar` there.

Secure Federated Search Between Releases 10.1.8 and 10.1.6

To set up secure federated search with a 10.1.8 instance as the federation broker and a 10.1.6 instance as the federation endpoint, consider the following:

- The federation broker and the federation endpoint must be connected to the same Oracle Internet Directory server.
- Federation parameters are not immediately updated. To see changes immediately, bounce the middle tier on the federation broker.
- If you are setting SSO mode 2 (private content alone protected by SSO) in the federation endpoint instance and you are not seeing private results returned by the federation broker instance, then you are hitting a 10.1.6 bug.

  Workaround: Open the `web.xml` file in `$ORACLE_HOME/oc4j/j2ee/oc4j_applications/applications/search_query/query/WEB-INF/web.xml`. Comment out the `filter` and `filter-mapping` elements:

```
<!-- commenting filter and filter-mapping due to bug 5072567
  <filter>
   <filter-name>RequestFilter</filter-name>
   <filter-class>oracle.search.query.RequestFilter</filter-class>
 </filter>

 <filter-mapping>
  <filter-name>RequestFilter</filter-name>
  <servlet-name>OracleSearch</servlet-name>
 </filter-mapping>
 -->
```

Then restart the middle tier with `searchctl restart`.

> **Note:** If you must have a 10.1.6 instance as the federation endpoint behind SSO, then you cannot configure the instance in secure mode 3.

- When using the endpoint application entity as the federation endpoint for creating the federated source, make sure to add this entity to the trusted application's group under the federation endpoint instance's application entity entry in Oracle Internet Directory. See the following section.

Oracle SES 10.1.8 federating to Oracle SES 10.1.6:

If the federation broker is Oracle SES 10.1.8 and the federation endpoint is Oracle SES 10.1.6, then the administrator of the broker instance must perform the following steps:

1. Get an entity name(DN) and password that is an entity under the trusted application's group of the application entity created for the Oracle SES 10.1.6 instance in Oracle Internet Directory. If there is no entity found in the trusted application's group, then either create a new entity or add the same application entity(DN) to the `uniqueMember` attribute of the endpoint's application entity. For example, if the application entity for the endpoint instance is:

   ```
   orclApplicationCommonName=oesEntity_
   endpoint,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
   ```

   add:

   ```
   orclApplicationCommonName=oesEntity_
   endpoint,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
   ```

   to the `uniqueMember` attribute of

   ```
   orclApplicationCommonName=oesEntity_
   endpoint,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
   ```

   If you are using the application entity of the 10.1.6 instance as the trusted entity, then the password for this entity is same as the Oracle SES admin password when Oracle SES was connected to the directory.

2. Create a federated source, and use the trusted entity created in the previous step for the **Remote Entity Name** and **Remote Entity Password**. The **Search User Attribute** should be the name of the attribute (in the directory to which broker is connected) corresponding to the `orclguid` attribute (in the Oracle Internet Directory the endpoint instance is connected to). If both broker and endpoint instance are connected to same Oracle Internet Directory, then the name of the attribute is `orclguid`.

Oracle SES 10.1.6 federating to Oracle SES 10.1.8:

If the federation broker is Oracle SES 10.1.6 and federation endpoint is Oracle SES 10.1.8, then the administrator of the endpoint instance must perform the following steps:

1. Get an entity name(DN) and password that is an entity under the trusted application's group of the application entity created for the SES 10.1.6 instance in Oracle Internet Directory. If there is no entity found in the trusted application's group, then either create a new entity or add the same application entity(DN) to the `uniqueMember` attribute of the endpoint's application entity. For example, if the application entity for the endpoint instance is:

   ```
   orclApplicationCommonName=oesEntity_
   endpoint,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
   ```

   add:

   ```
   orclApplicationCommonName=oesEntity_
   broker,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
   ```

   to the `uniqueMember` attribute of

```
orclApplicationCommonName=oesEntity_
broker,cn=OES,cn=Products,cn=OracleContext,dc=us,dc=oracle,dc=com
```

If you are using the application entity of the 10.1.6 instance as the trusted entity, then the password for this entity is same as the Oracle SES admin password when Oracle SES was connected to the directory.

2. Create a federation trusted entity on the broker instance with the entity name and password obtained from the previous step. This should be the name of the attribute (in the directory to which endpoint is connected) corresponding to the `orclguid` attribute (in the Oracle Internet Directory the broker instance is connected to). If both broker and endpoint instance are connected to same Oracle Internet Directory, then the name of the attribute is `orclguid`.

B

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
962	Document was excluded based on mime type	STATUS_DOC_MIME_TYPE_ EXCLUDED	30041
964	Document was excluded based on boundary rules	STATUS_DOC_BOUNDARY_ RULE_EXCLUDED	30258
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

C

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 in 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 re-crawl.
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.

D

WSDL Specifications

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 descriptions of the Oracle SES Web Services APIs:

- Query Web Service API
- Admin Web Service API

See Also: "Oracle Secure Enterprise Search Web Services APIs" on page 7-2

Query Web Service API

```
<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:complexType name="SCElement">
        <xsd:all>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="content" type="xsd:base64Binary"/>
        </xsd:all>
      </xsd:complexType>

      <xsd:complexType name="SCElementArray">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:SCElement[]"/>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>

  </types>



  <!-- Messages for Oracle Secure Enterprise Search Web Service 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="doOracleAdvancedSearch">
  <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"/>
  <part name="searchControls"     type="xsd:string"/>
</message>

<message name="doOracleAdvancedSearchResponse">
  <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>

<message name="getSuggestedContent">
  <part name="query"        type="xsd:string"/>
  <part name="returnType"        type="xsd:string"/>
</message>

<message name="getSuggestedContentResponse">
  <part name="return"         type="typens:SCElementArray"/>
</message>

<!-- Port for Oracle SES Web Service 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="doOracleAdvancedSearch">
      <input message="tns:doOracleAdvancedSearch"/>
      <output message="tns:doOracleAdvancedSearchResponse"/>
    </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>

    <operation name="getSuggestedContent">
<input message="tns:getSuggestedContent"/>
<output message="tns:getSuggestedContentResponse"/>
    </operation>

  </portType>

  <!-- Binding for Oracle SES Web Service 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="doOracleAdvancedSearch">
      <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>

    <operation name="getSuggestedContent">
<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 SES Web Service APIs -->
  <service name="OracleSearchService">
    <port name="OracleSearchPort" binding="tns:OracleSearchBinding">
      <soap:address location="http://myserver:7777/search/query/OracleSearch"/>
    </port>
  </service>

</definitions>
```

Admin Web Service API

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions
     name="OracleSearchAdminService"
     targetNamespace="http://search.oracle.com/AdminService/2006-09-15"
     xmlns="http://schemas.xmlsoap.org/wsdl/"
     xmlns:tns="http://search.oracle.com/AdminService/2006-09-15"
     xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
     xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    >
    <types>
        <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://search.oracle.com/AdminService/2006-09-15"
             elementFormDefault="qualified"
xmlns:tns="http://search.oracle.com/AdminService/2006-09-15"
             xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/">
            <complexType name="getEstimatedIndexFragmentation">
                <sequence/>
            </complexType>
            <complexType name="getEstimatedIndexFragmentationResponse">
                <sequence>
                    <element name="result" type="int"/>
                </sequence>
            </complexType>
            <complexType name="getScheduleStatus">
                <sequence>
                    <element name="name" type="string" nillable="true"/>
                    <element name="locale" type="string" nillable="true"/>
                </sequence>
            </complexType>
            <complexType name="getScheduleStatusResponse">
                <sequence>
                    <element name="result" type="tns:ScheduleStatus"
nillable="true"/>
                </sequence>
            </complexType>
            <complexType name="ScheduleStatus">
                <sequence>
                    <element name="nextCrawl" type="dateTime" nillable="true"/>
                    <element name="status" type="string" nillable="true"/>
                    <element name="lastCrawled" type="dateTime" nillable="true"/>
                    <element name="translatedStatus" type="string"
nillable="true"/>
                    <element name="errorLog" type="string" nillable="true"/>
                </sequence>
            </complexType>
            <complexType name="getSchedules">
                <sequence>
                    <element name="locale" type="string" nillable="true"/>
                </sequence>
            </complexType>
            <complexType name="getSchedulesResponse">
                <sequence>
                    <element name="result" type="tns:Schedule" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
```

```
            </sequence>
        </complexType>
        <complexType name="Schedule">
            <sequence>
                <element name="currentStatus" type="tns:ScheduleStatus"
nillable="true"/>
                <element name="assignedSources" type="string" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="name" type="string" nillable="true"/>
            </sequence>
        </complexType>
        <complexType name="login">
            <sequence>
                <element name="username" type="string" nillable="true"/>
                <element name="pwd" type="string" nillable="true"/>
            </sequence>
        </complexType>
        <complexType name="loginResponse">
            <sequence/>
        </complexType>
        <complexType name="logout">
            <sequence/>
        </complexType>
        <complexType name="logoutResponse">
            <sequence/>
        </complexType>
        <complexType name="optimizeIndexNow">
            <sequence/>
        </complexType>
        <complexType name="optimizeIndexNowResponse">
            <sequence/>
        </complexType>
        <complexType name="startSchedule">
            <sequence>
                <element name="name" type="string" nillable="true"/>
            </sequence>
        </complexType>
        <complexType name="startScheduleResponse">
            <sequence/>
        </complexType>
        <complexType name="stopSchedule">
            <sequence>
                <element name="name" type="string" nillable="true"/>
            </sequence>
        </complexType>
        <complexType name="stopScheduleResponse">
            <sequence/>
        </complexType>
        <element name="getEstimatedIndexFragmentationElement"
type="tns:getEstimatedIndexFragmentation"/>
        <element name="getEstimatedIndexFragmentationResponseElement"
type="tns:getEstimatedIndexFragmentationResponse"/>
        <element name="getScheduleStatusElement"
type="tns:getScheduleStatus"/>
        <element name="getScheduleStatusResponseElement"
type="tns:getScheduleStatusResponse"/>
        <element name="getSchedulesElement" type="tns:getSchedules"/>
        <element name="getSchedulesResponseElement"
type="tns:getSchedulesResponse"/>
        <element name="loginElement" type="tns:login"/>
```

```
            <element name="loginResponseElement" type="tns:loginResponse"/>
            <element name="logoutElement" type="tns:logout"/>
            <element name="logoutResponseElement" type="tns:logoutResponse"/>
            <element name="optimizeIndexNowElement" type="tns:optimizeIndexNow"/>
            <element name="optimizeIndexNowResponseElement"
type="tns:optimizeIndexNowResponse"/>
            <element name="startScheduleElement" type="tns:startSchedule"/>
            <element name="startScheduleResponseElement"
type="tns:startScheduleResponse"/>
            <element name="stopScheduleElement" type="tns:stopSchedule"/>
            <element name="stopScheduleResponseElement"
type="tns:stopScheduleResponse"/>
        </schema>
    </types>
    <message name="OracleSearchAdminService_getEstimatedIndexFragmentation">
        <part name="parameters"
element="tns:getEstimatedIndexFragmentationElement"/>
    </message>
    <message name="OracleSearchAdminService_
getEstimatedIndexFragmentationResponse">
        <part name="parameters"
element="tns:getEstimatedIndexFragmentationResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_getScheduleStatus">
        <part name="parameters" element="tns:getScheduleStatusElement"/>
    </message>
    <message name="OracleSearchAdminService_getScheduleStatusResponse">
        <part name="parameters" element="tns:getScheduleStatusResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_getSchedules">
        <part name="parameters" element="tns:getSchedulesElement"/>
    </message>
    <message name="OracleSearchAdminService_getSchedulesResponse">
        <part name="parameters" element="tns:getSchedulesResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_login">
        <part name="parameters" element="tns:loginElement"/>
    </message>
    <message name="OracleSearchAdminService_loginResponse">
        <part name="parameters" element="tns:loginResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_logout">
        <part name="parameters" element="tns:logoutElement"/>
    </message>
    <message name="OracleSearchAdminService_logoutResponse">
        <part name="parameters" element="tns:logoutResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_optimizeIndexNow">
        <part name="parameters" element="tns:optimizeIndexNowElement"/>
    </message>
    <message name="OracleSearchAdminService_optimizeIndexNowResponse">
        <part name="parameters" element="tns:optimizeIndexNowResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_startSchedule">
        <part name="parameters" element="tns:startScheduleElement"/>
    </message>
    <message name="OracleSearchAdminService_startScheduleResponse">
        <part name="parameters" element="tns:startScheduleResponseElement"/>
    </message>
    <message name="OracleSearchAdminService_stopSchedule">
```

```
        <part name="parameters" element="tns:stopScheduleElement"/>
    </message>
    <message name="OracleSearchAdminService_stopScheduleResponse">
        <part name="parameters" element="tns:stopScheduleResponseElement"/>
    </message>
    <portType name="OracleSearchAdmin">
        <operation name="getEstimatedIndexFragmentation">
            <input message="tns:OracleSearchAdminService_
getEstimatedIndexFragmentation"/>
            <output message="tns:OracleSearchAdminService_
getEstimatedIndexFragmentationResponse"/>
        </operation>
        <operation name="getScheduleStatus">
            <input message="tns:OracleSearchAdminService_getScheduleStatus"/>
            <output message="tns:OracleSearchAdminService_
getScheduleStatusResponse"/>
        </operation>
        <operation name="getSchedules">
            <input message="tns:OracleSearchAdminService_getSchedules"/>
            <output message="tns:OracleSearchAdminService_getSchedulesResponse"/>
        </operation>
        <operation name="login">
            <input message="tns:OracleSearchAdminService_login"/>
            <output message="tns:OracleSearchAdminService_loginResponse"/>
        </operation>
        <operation name="logout">
            <input message="tns:OracleSearchAdminService_logout"/>
            <output message="tns:OracleSearchAdminService_logoutResponse"/>
        </operation>
        <operation name="optimizeIndexNow">
            <input message="tns:OracleSearchAdminService_optimizeIndexNow"/>
            <output message="tns:OracleSearchAdminService_
optimizeIndexNowResponse"/>
        </operation>
        <operation name="startSchedule">
            <input message="tns:OracleSearchAdminService_startSchedule"/>
            <output message="tns:OracleSearchAdminService_startScheduleResponse"/>
        </operation>
        <operation name="stopSchedule">
            <input message="tns:OracleSearchAdminService_stopSchedule"/>
            <output message="tns:OracleSearchAdminService_stopScheduleResponse"/>
        </operation>
    </portType>
    <binding name="SearchAdminSoapBinding" type="tns:OracleSearchAdmin">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="getEstimatedIndexFragmentation">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/getEstimatedIndexFrag
mentation"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="getScheduleStatus">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/getScheduleStatus"/>
```

```
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="getSchedules">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/getSchedules"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="login">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/login"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="logout">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/logout"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="optimizeIndexNow">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/optimizeIndexNow"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="startSchedule">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/startSchedule"/>
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
        <operation name="stopSchedule">
            <soap:operation
soapAction="http://search.oracle.com/AdminService/2006-09-15/stopSchedule"/>
```

```
            <input>
                <soap:body use="literal" parts="parameters"/>
            </input>
            <output>
                <soap:body use="literal" parts="parameters"/>
            </output>
        </operation>
    </binding>
    <service name="OracleSearchAdminService">
        <port name="SearchAdmin" binding="tns:SearchAdminSoapBinding">
            <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
        </port>
    </service>
</definitions>
```

E

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 5-27 and "Setting Up Secure Oracle Content Database Sources" on page 5-28

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 Encryption
dn: cn=PwdPolicyEntry,cn=Common,cn=Products,cn=OracleContext
changetype: modify
replace: orclpwdencryptionenable
orclpwdencryptionenable: 1
```

F

Third Party Licenses

This appendix includes the third party license for all the third party products included with Oracle Secure Enterprise Search. This appendix includes the following topics:

Apache Software

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.

```
                 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 CONDITION

Plug-in Software

Oracle SES ships a several *plug-ins* to enterprise sources. (Plug-ins allow Oracle SES to crawl and index content in proprietary systems, such as Siebel). For some plug-ins to work, additional software may need to be installed and licensed from the respective vendor; for example, EMC Documentum requires Documentum Foundation Classes (DFC), a Java library, to be installed on the machine running Oracle SES.

The following enterprise sources require additional software to be installed on the machine running Oracle SES:

- EMC Documentum Content Server
- FileNet Content Engine and FileNet Image Server
- Open Text
- Microsoft Exchange
- Microsoft NTFS may require Microsoft's .NET 2.0

See Also: Chapter 5, "Configuring Access to Enterprise Content Sources" for detailed information about each source type

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

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. A *federation broker* calls the *federation endpoint* to collect content matching the search criteria for the sources managed at that endpoint.

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

R

S

T

U

W