Implementing a Custom Search Interface with SES - a case study with search.oracle.com

An Oracle White Paper
June 2006

ORACLE

Implementing a Custom Search Interface with SES - a case study with search.oracle.com

Implementing a Custom Search Interface with SES - a case study with search.oracle.com

INTRODUCTION

Oracle Secure Enterprise Search 10g is an out-of-the-box solution that provides search capabilities across multiple repositories - Oracle databases and Portals, websites, email systems, files on disk and many more.
This paper provides an outline on using Oracle Secure Enterprise Search's web service APIs for developing a custom UI, based on the experience of its implementation. The goal of this document is to understand how the search user interface implementation using web service APIs of SES works with oracle.com. The UI can be viewed at search.oracle.com. Even though Oracle Secure Enterprise Search provides a default search UI, search.oracle.com needed a custom UI, because OTN has its own look and feel and, in order to make the UI consistent, the search engine has to have a custom UI. For more details about Oracle Secure Enterprise Search and sample code, visit
http://www.oracle.com/technology/products/oses/index.html

This implementation uses freemarker as the template engine to generate text output for the interface.

CUSTOM SEARCH UI

The Web Services API is used to build a custom search application to search on the data indexed by Oracle Secure Enterprise Search. Oracle SES provides the Web Services Proxy Java library.

HIGH LEVEL DESIGN

The custom UI relies on the Java client stubs generated from the SES WSDL. The approach in this implementation is to separate the UI from scriplets by using freemarker as the template technology. This provides a reasonable separation of the application logic from the presentation layer.

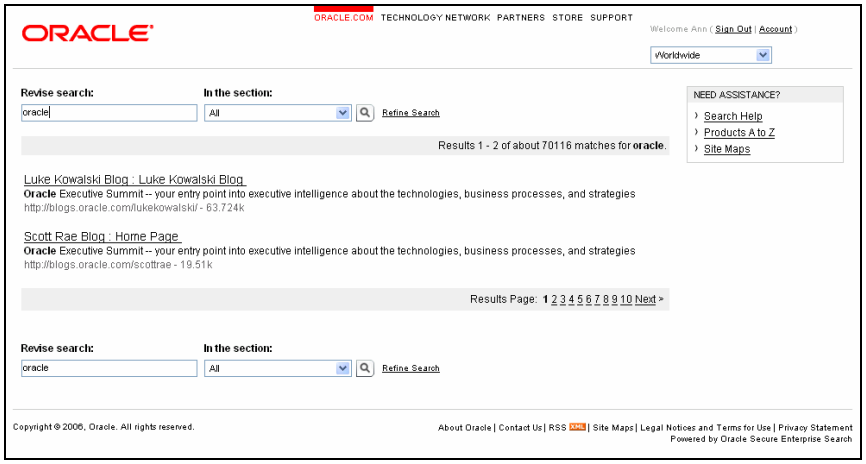
In this implementation, there are two jsps such as initialize.jsp and search.jsp. initialize.jsp initialize the search context and set the web service end point. The search.jsp, which is the entry point, will process the search query and hold all the dynamic content, which are required for the proper display of UI, in a hashmap. There are three different templates designed for custom search UI such as query.ftl,

result.ftl and noresult.ftl. query.ftl is the template corresponding to the default search UI. Depending upon the search result for the query submitted in the default search UI, application will load a template, either result.ftl or noresult.ftl, which is actually responsible for the UI look and feel. Result UI, which gets generated from result.ftl, will have the search results. When there is no search result for a keyword submitted, the noresult screen, which is generated by noresult.ftl, will get displayed.

Custom Search UI Snapshots - search.oracle.com

Query screen

Result Screen



No Result Screen

Implementation of custom UI using web services

The Web Services API is used to build custom search application to search on the data indexed by Oracle Secure Enterprise Search.

In order to get the complete picture of web service API's provided by SES, refer Oracle® Secure Enterprise Search Administrator's Guide.

Web service end point

The default end point for accessing web service APIs for SES is

http://hostname:port/search/query/OracleSearch. How to change the default end point is described in the section "Deploying the Search Custom UI".

Note: After changing the default namespace to /default/search the end point will be http://hostname:port/default/query/OracleSearch.

Package to import

The following package needs to be imported to implement OSES web service APIs.

```
oracle.search.query.webservice.client.*;
```

Initialize Search Context

The search context can be initialized as follows.

```
OracleSearchService ctx = new OracleSearchService();

ctx.setSoapURL( <endpoint> );
```

e.g.

```
ctx.setSoapURL(http://hostname:port/default/query/Orac
leSearch);
```

Search

Search can be performed within group as well as within a node. If none of this specified, search would be performed against all available sources configured in the SES admin UI. If search has to be specified within a group, the group name has to be specified. Similarly for search within a node, the nodeid and federated Id are to be specified. For a local search, the fid is –1 by default. For node search within federated source, the corresponding fid has to be given as input.

Data groups

SES provides API to get the configured source groups. This information can be used display the available groups in the UI and restrict search within a source group.

While using federated search, the source can be added to a group and using this group search can be restricted within the federated source.

Attributes

SES provides API to get the attributes available for search. This attribute list can be used for advanced search where any passed attribute can be validated against the attributes available. This is useful for narrow down the search by restricting the match against any specific attribute.

Filters

Filters are used to restrict the search query. While doing a search, multiple filters can be specified. Each filter is a restriction on search results. Filters are connected by filterConnector. No filter applies to the search result if not set explicitly.

Filter can be created as follows:

```
Filter filter = new Filter(attributeId,
attributetype,searchOpName,  searchAttValue);
```

AttributeId and attributeType can be obtained by matching the attribute name from the attributeList obtained using API getAttributes.

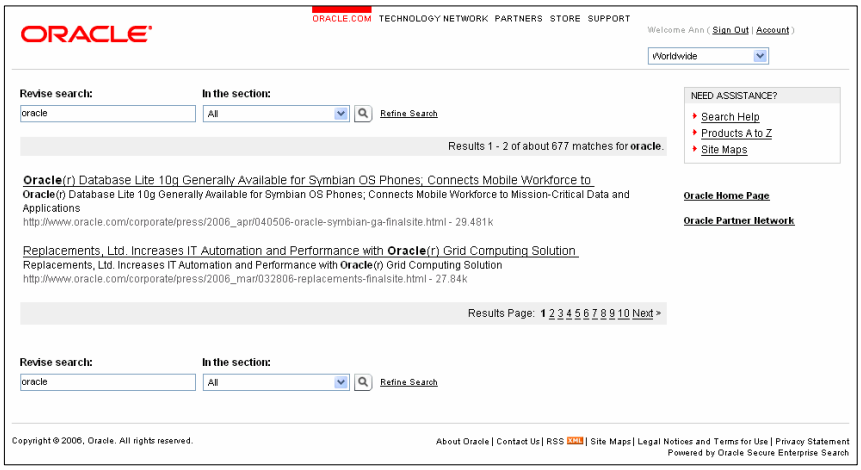
For e.g. if the attributeId and type of an attribute LastModifiedDate are 1 and Date, and the operation name and attributevalue are greaterthan and 10/10/2005, the object will be created as

```
Filter filter = new Filter(1, “Date”, “greaterthan”,
“10/10/2005”);
```

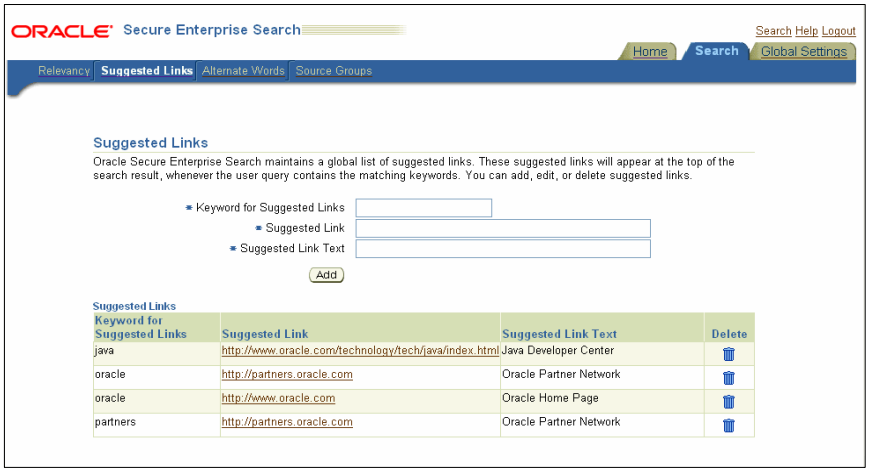
Suggested Links

Suggested Links is a feature which improves the overall search quality. For example, for a keyword oracle, suggested links like http://www.oracle.com and http://partners.oracle.com can be added so that end user will get this also with the result for keyword oracle and this can be shown with high priority on the UI. Suggested links for any search keyword can be added in the OSES admin UI. This

information can be shown in the custom UI if required. The OracleSearchResult object, which is returned by the search API hold an array of Suggested Link class objects. It has the suggested link title and url as member variables. A screen shot of the Custom UI with Suggested Links can be viewed in below



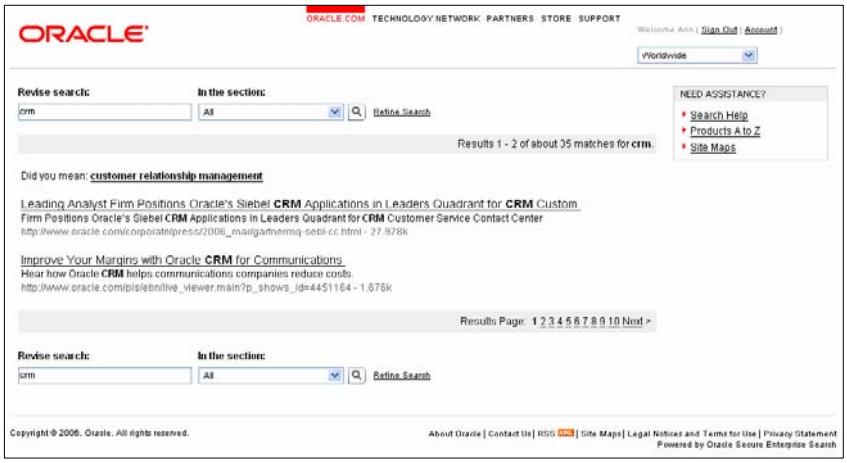
The snapshot below shows the page in admin UI where the suggested links can be added.



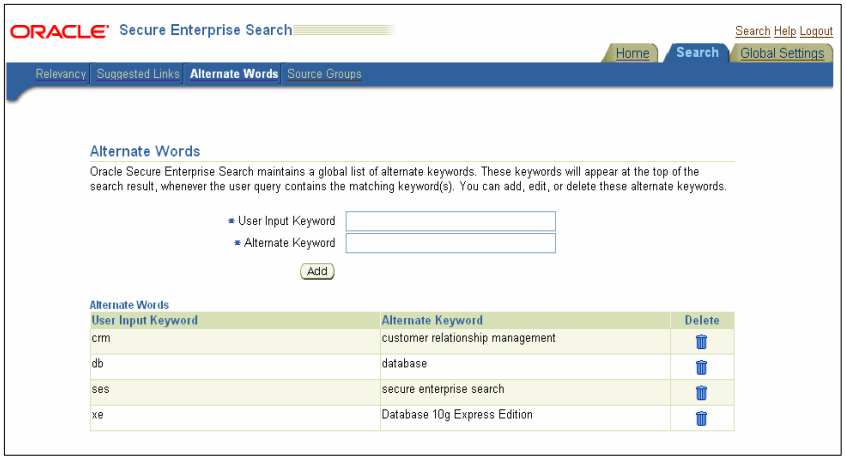
Alternate Keyword

Alternate Keyword is another feature, which improves the overall search quality. Alternate Keywords are used to search commonly used keywords, so that the user can easily specify the exact keyword he was intended to search. For example if there is an alternate keyword mentioned in admin UI for a keyword "crm" as

"customer relationship management", when the user search for "crm", the result page show a link *Did you mean* "customer relationship management", clicking on which will yield the results for "customer relationship management". The alternate keyword for any search keyword can be configured in the OSES admin UI. This can be displayed in the UI if required. The OracleSearchResult object, which is returned by the search API has a member variable holding the alternate keywords. A screen shot of the Custom UI with Alternate Keyword can be viewed below.



The snapshot below shows the page in admin UI where the Alternate Keywords can be added.



Similar Documents

All duplicate and near duplicate documents comes in the search result are referred by the term similar documents. If in the search result, duplicate documents need not come, then the search query should be made in such a way that dupRemoved flag should be set to true. If the similar documents are to be there in the search result, dupRemoved flag should be set to false and dupMarked flag should be set to true. If the requirement is that the similar documents should get displayed only by

clicking a link say Similar Documents, code should be capable of hiding the duplicate documents initially, based on the isDuplicate flag value, and display it on clicking the link, by passing the document id.

Search Result

The search results are returned as an object of OracleSearchResult.

The OracleSearchResult object has an array of ResultElement objects.

Any of its member variable values can be displayed as per the business requirement.

Secure Enterprise search

Secure enterprise can be implemented using the authentication APIs. In order to use the authentication APIs, the custom UI has to keep the end user's requests in the same context

```
ctx = (OracleSearchService)
request.getSession().getAttribute("SearchContext");

if (ctx == null)

{

      ctx = new OracleSearchService();

      ctx.setSoapURL( endpoint );
      request.getSession().setAttribute("SearchContext
      ",ctx);

}
```

Packaging for Deployment

Classpath Setting

To compile and run client application using Oracle SES client-side Java proxy libraries, one need to include the following files in the Java CLASSPATH. These files can be obtained from Oracle SES server file directory.

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

The application is packaged within search.ear. Deployment of this ear within oses does not require restarting of oc4j.

Configuration

The web service end point is configured in web.xml. In order to deploy the same ear under different ses, orion-web.xml can be used to override the entries mentioned in web.xml.

The location of the freemarker templates is also configurable. The path can be outside ear. This makes it easy to change the UI without a redeployment of the ear. The implementation is in such a way that, if the templates are not present in the configured path, it will be taken up from the ear.

Deploying the Search Custom UI

SES default namespace

The default SES SEARCH UI has the name space /search/search. In order to use the same name space for the custom UI, the namespace for default UI has to be changed. In the implementation described here, the default search UI namespace has been changed to /default/search so that /search/search can be used by custom UI.

Note: If the custom UI is being deployed in a different oc4j instance than OSES, there is no need of changing the namespace.

How to change default name space is described below

Creating Separate Namespace

1. Changes to the http-web-site.xml for the namespace change

   Backup file http-web-site.xml at

   `$ORACLE_HOME/oc4j/j2ee/OC4J_SEARCH/config/`

   Original:

```
<default-web-app application="search_query"
name="welcome"/>
<web-app application="search_query" name="query"
load-on-startup="true" root="/search/query" />
<web-app application="search_admin" name="admin"
load-on-startup="true" root="/search/admin" />
<web-app application="search_admin" name="ohw"
load-on-startup="true" root="/search/ohw" />
<web-app application="monitor" name="monitor" load-
on-startup="true" root="/monitor" />
```

Add/Change the lines shown in bold below so that the applications listed include the root for the default query and the new namespace query as follows:

```
<default-web-app application="search_query"
name="welcome"/>
<web-app application="search_query" name="query"
load-on-startup="true" root="/default/query" />
<web-app application="search_admin" name="admin"
load-on-startup="true" root="/search/admin" />
<web-app application="search_admin" name="ohw"
load-on-startup="true" root="/search/ohw" />
<web-app application="monitor" name="monitor" load-
on-startup="true" root="/monitor" />
<web-app application="search" name="search" load-
on-startup="true" root="/search" />
```

2. Back up and modify index.jsp at `$ORACLE_HOME/oc4j/j2ee/home/default-web-app/.` The new file will have this entry only:

   ```
   <%
   response.sendRedirect(response.encodeRedirectUrl
   ("/search/search")); %>
   ```

   Purpose of this change is so that if someone types http://hostname:port, he is automatically redirected to custom UI and not to default search UI.

3. Back up index.html at `$ORACLE_HOME/oc4j/j2ee/oc4j_applications/applications/search_query/welcome/`

   Create a new index.html file with the following entry:

   ```
   <head>
   <meta http-equiv="refresh"
   content="0;url=/search/search"/>
   </head>
   ```

Purpose of this change is so that if someone types http://hostname:port, he/she is automatically redirected to custom UI and not to default search UI.

CONCLUSION

To develop a custom UI as per the business requirements is a necessity. Oracle Secure Enterprise Search Web Services API lets you write your own application to query Oracle SES over the network. It is certified to work with Java. The API provides the following benefits:

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

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

APPENDIX

Freemarker Implementation

The Custom UI implementation described here uses FreeMarker for UI implementation. FreeMarker is a "template engine", a generic tool to generate output based on templates. The template separates the actual UI definition from the jsp so that changing the template will not result in an ear redeployment. The dynamic content to be displayed in the UI will be stored in a hashmap within the jsp and the static part will be within the template. The dynamic part will be associated to the template and can be accessed in the template by using the name by which it is stored within the hashmap such as ${<name>}.

There can be a setting, which can be configured to set the time in seconds, which needs to elapse before freemarker checks for a newer version of the template. If this value is set to 0, every time when the template is loaded, freemarker will check for a newer version and any change made will be immediately effected.

Details of freemarker can be found at http://freemarker.sourceforge.net/.

Custom UI Implementation using web services
June 2006
Author: Ann Jacob
Contributing Authors: Sudhir Dureja

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com