

Oracle  
Ultrasearch  
Configuration  
And  
Usage

**Author:** G S Chapman  
**Date:** 9<sup>th</sup> March 2005  
**Version:** 1.1  
**Location of Document:**

## DOCUMENT HISTORY

Version	Date	Changed By:	Remarks
	09/03/2005	G S Chapman	Initial Version

## DOCUMENT DISTRIBUTION

Copy No	Name	Role	Organisation

## **DOCUMENT REFERENCES**

<b>Document Name</b>	<b>Originator</b>	<b>Part Number</b>	<b>Version</b>	<b>Date</b>

## **TABLE OF CONTENTS**

<b>1</b>	<b>Installation .....</b>	<b>1</b>
1.1	Instance Setup .....	1
1.2	Configuration of Middle tier Ultrasearch instance.....	1
1.3	Crawler Setup.....	3
1.4	Web Setup .....	4
1.5	Attributes.....	4
1.6	Sources .....	4
1.7	Schedules .....	5
1.8	Globalization .....	5
1.9	Application setting .....	5
<b>2</b>	<b>Screen Setups .....</b>	<b>6</b>
2.1	Primary Screen .....	6
2.2	Advanced Screen .....	6

## **TABLE OF FIGURES**

Figure 1 - Default Ultrasearch screen.....	6
Figure 2 - Default advanced Ultrasearch screen .....	6

## **PURPOSE OF DOCUMENT**

This document briefly describes the setup and usage of Oracle Ultrasearch as supplied in the standard database 'out of the box'. It also includes a sample JSP piece of code for changing the default screen. Based upon Oracle 9.2.

# Ultrasearch

*Note: This is very specific to Application Server 9.0.2. For Ultrasearch in a standard database, or for a different version of the Application Server the files and their locations can and will be different.*

## 1 INSTALLATION

Ultrasearch is installed as part of the standard Oracle product installation. Once installed a few configuration steps need to be performed. This particular setup does not have a requirement does not require replication of the Ultrasearch results between different application servers. For this reason the Ultrasearch instance is located within the Infrastructure database upon the Application Server. All of the following configuration and setup steps are therefore performed against the default Infrastructure database instance installed on the Application Server.

### 1.1 Instance Setup

As released the databases are not configured for heavy Ultrasearch usage. The user CTXSYS will require a larger temporary tablespace, or a separate temporary tablespace for its own use. It is recommended that a separate temporary tablespace of at least 50M is used for the CTXSYS user.

Each Ultrasearch instance owner should also have it's own separate tablespace. For the test installation this has been left with the default 'USERS' tablespace since there will be no other usage by other users of this tablespace and there will only be one Ultrasearch instance owner. (i.e. SCOTT). The test configuration imade use of the default Temporary tablespace for the Ultrasearch instance owner (Scott).

Every Ultra Search instance is associated with one and only one database schema. To create a new Ultra Search instance, associate the new instance with an existing database schema. The database schema must not already be associated with another Ultra Search instance. For our setup a particular user already present with the Infrastructure database was chosen. This user 'SCOTT' was given the necessary permission to enable usage of Ultrasearch. These permissions are:

```
GRANT WKUSER to SCOTT;
```

This does not allow SCOTT to be an Ultrasearch administrator for which a different role required.

Regular running of statistics on tables wk\$url, wk\$doc and dr\$wk\$doc are required to avoid performance issues.

By default the Ultrasearch administrator is a user WKSYS which is locked with a randomised password within the database when the database instance is configured. This user needs to be unlocked and the password changed to a suitable value. Once so changed, the WKSYS user can log in and perform the creation and administration of Ultrasearch instances.

### 1.2 Configuration of Middle tier Ultrasearch instance.

In the \$ORACLE\_HOME/j2ee/OC4J\_Portal/config directory, edit the file data-sources.xml. Under the <data-sources> tag add the following:

```
<data-source
```

```
class="oracle.jdbc.pool.OracleConnectionCacheImpl"  
name="UltraSearchDS"  
location="jdbc/UltraSearchPooledDS"  
username="scott"  
password="tiger"  
url="jdbc:oracle:thin:@apps.server.addr:1521:iasdb"  
</>
```

Edit the ultrasearch.properties file. The ultrasearch.properties file specifies the database the Web application and JSP application connects to. This file is located in the directory \$ORACLE\_HOME/ultrasearch/webapp/config.

Edit the line that begins with connection URL. Also add the admin.srchome line for the system. Something like the following is required.

```
connection.driver=oracle.jdbc.driver.OracleDriver  
connection.url=jdbc:oracle:thin:@apps.server.addr:1521:iasdb  
admin.srchome=/home/oracle/product/ora9ias_apps/ultrasearch/webapp/isearch_admin
```

Once the above changes have been made the web server (Apache) should be stopped and restarted. To test if the HTTP server is started visit the UltraSearch welcome page:

<http://hostname:7778/ultrasearch/index.html>

The installation and configuration can be tested by using the samples supplied by Oracle. Tests including making a connection to the Ultrasearch administration screen, and configuring a test instance, and also using the sample test Ultrasearch script to view the results for the test instance.

The Ultrasearch Administration screen can be reached at the following URL:

<http://hostname:7778/ultrasearch/admin/control/login.jsp>

If a blank page is encountered when you have logged into the admin tool, then the value of the admin.srchome in the ultrasearch.properties file is probably wrong. Try changing the line such that there are double back slashes in the line.

i.e. admin.srchome=C:\\oracle\\ora\\ultrasearch\\webapp\\isearch\_admin

If this does not fix your problem consult your database administrator.

*[There are many possible causes for the blank screen, one that has been encountered previously is caused by the entries in the \$ORACLE\_HOME/j2ee/OC4J\_Portal/config/application.xml file being incorrectly populated or even not populated at all.]*

Once Ultra Search has been successfully installed, the database and listener is running where Ultra Search are configured. You are able to successfully login to the Ultra Search Administration Tool, crawl and index a data source successfully, however, the sample Ultra Search JSP Query Application will need configuring to enable you to login successfully.

The sample jsp files are located in the ORACLE\_HOME/ultrasearch/sample/query/9i directory. The sample JSP pages (gsearch.jsp, gsearchf.jsp, display.jsp, mail.jsp, gutil.jsp) all require the user to edit the file and change the database login username/password. They are shipped to login as wk\_test/wk\_test, you should change that to the instance owner's login parameters.

```
qt.SetUser("wk_test");  
qt.SetPassword("wk_test");
```

Once configured the sample search screen can be used. The sample Ultrasearch query screen can be reached at the following URL:

<http://hostname:7778/ultrasearch/query/search.jsp>

## 1.3 Crawler Setup

**Crawler Threads:** Specify the number of crawler threads to be spawned at run time.

**Number of Processors:** Specify the number of central processing units (CPUs) that exist on the server where the Ultra Search crawler will run. This setting determines the optimal number of document conversion threads used by the system. A document conversion thread converts multi-format documents into HTML documents for proper indexing. A value of 20 has been used successfully in our testing.

**Crawling Depth:** A Web document could contain links to other Web documents, which could contain more links. This setting lets you specify the maximum number of nested links the crawler will follow. A depth of 4 has been found suitable for most sites.

**Crawler Timeout Threshold:** Specify in seconds a crawler timeout. The crawler timeout threshold is used to force a timeout when the crawler cannot access a Web page. 30 seconds is a reasonable value but this may need changing in the light of experience with some of the partner sites. *[An initial test on one site required a value of several minutes to get any pages for indexing.]*

**Default Character Set:** Specify the default character set. The crawler uses this setting when an HTML document does not have its character set specified.

**Temporary Directory Location and Size:** Specify a temporary directory and size. The crawler uses the temporary directory for intermittent storage during indexing. Specify the absolute path of the temporary directory. The size is the maximum temporary space in megabytes that will be used by the crawler. The size of the temporary directory is important because it affects index fragmentation. The smaller the size, the more fragmented the index. As a result, the query will be slower, and index optimization needs to be performed more frequently. Increasing the directory size reduces index fragmentation, but it also reduces crawling throughput (total number of documents crawled each hour). This is because it takes longer to index a bigger temporary directory, and the crawler needs to wait for the indexing to complete before it can continue writing new documents to the directory.

**Choose:** For Solaris this would be `'/tmp'`, or for a Windows server `'C:\TEMP'` as a suggestion. For size 25M is a good starting point.

**Crawler Logging:** Specify the following:

**Level of detail:** everything or only a summary: Choose `'Summary only'`

**Crawler logfile directory:** Choose `'/tmp'` if Solaris OR `'C:\TEMP'` if Windows.

**Crawler logfile language:** Choose `'English'`

The log file directory stores the crawler log files. 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. The crawler logfile language is the language the crawler uses to generate the log file.

**Database Connect String :** The database connect string is a standard JDBC connect string used by the remote crawler when it connects to the database. The connect string can be provided in the form of `[hostname]:[port]:[sid]` or in the form of a TNS keyword-value syntax. There is no connect string for a crawler launched locally on the server machine. Instead, it uses an Oracle JDBC OCI driver to connect to Oracle. The following is suggested:



apps.server.addr:1521:iasdb

Where iasdb is the database name, 1521 the default port and `apps.server.addr` is the fully qualified domain name of the application server.

## 1.4 Web Setup

No configuration or changes are necessarily required, but it will depend upon the site configuration.

## 1.5 Attributes

Search attributes are attributes exposed to the query user. Oracle Ultra Search provides system-defined attributes, such as "Author" and "Description." Ultra Search maintains a global list of search attributes. You can add, edit, or delete search attributes. You can also click Manage LOV to change the list of values (LOV) for the search attribute. There are two categories of attribute LOVs: one is global across all data sources, the other is data source-specific. To define your own attribute, enter the name of the attribute in the text box; select string, date, or number; and click Add. You can add or delete LOV entry and display name for search attributes. Display name is optional. If display name is absent, then LOV entry is used in the query screen.

Note: LOV is only represented as string type.

For our sample installation two attributes were added. These were Authority and Category.Subject. The first of these was set up with default LOV values. LOV values were also set up for the other partners, but unfortunately could not be tested since the relevant web sites are not tagged.

## 1.6 Sources

Only Web sources will be configured. A Web source represents HTML content on a specific Web site. Web sources differ from other data source types because they exist specifically to facilitate maintenance crawling of specific Web sites.

To create a new Web source, do the following:

- Specify a name for the Web source.
- Override the default crawler settings, such as default language, for each Web source. This step is optional. If you change any of the default settings, click Update. The defaults are suitable for our sample usage.
- Enter a starting address. This is the URL for the crawler to begin crawling. i.e. `http://www.macrotone.co.uk`
- Set URL boundary rules to refine the crawling space. You can include or exclude hosts and URL paths. An exclusion domain `uk.oracle.com` prevents the crawler from crawling Oracle hosts in the United Kingdom. Exclusion rules always override inclusion rules.
- Specify the types of documents the Ultra Search crawler should process for this source. HTML and plain text are default document types that the crawler will always process. No additional document type were specified for the test environment.
- Define, edit, or delete metatag mappings for your Web source. Metatags are descriptive tags in the HTML document header. One metatag can map to only one search attribute. For our test two specific metatype definitions were used. These reference the Dublin Core categorisation and were `DC.SUBJECT_CATEGORY` and `DC.COVERAGE.SPATIAL`. These were

mapped to two attributes created from the 'Attributes' tab. The second to an attribute called 'Authority' and the first to an attribute called 'Subject.Category'.

## 1.7 Schedules

This page is used to schedule data synchronization and index optimisation. Data synchronization means keeping the Ultra Search index up to date with all data sources. Index optimisation means keeping the updated index optimised for best query performance.

A synchronization schedule has one or more data sources assigned to it. The synchronization schedule frequency specifies when the assigned data sources are to be synchronized. Synchronization schedules are sorted first by name. Within a synchronization schedule, individual data sources are listed and can be sorted by source name or source type.

Our schedules were set up for each of the partner sites to run at a desired frequency. In each case the web source for the schedule was defined, such that each schedule applied to an individual partner's site. In our test environment the schedules once tested were disabled to prevent unnecessary running.

## 1.8 Globalization

Ultra Search lets you translate names to different languages. This page lets you enter multiple values for search attributes, LOV display names, and data groups. This section is used to change the titles of the fields displayed in the web page windows.

**Search Attribute Name:** This section lets you translate attribute display names to different languages.

**LOV Display Name:** This section lets you translate data group names to different languages.

**Data Group Name:** This section lets you translate data group display names to different languages.

The Globalisation section is being used to change the text that is displayed for the Search attributes only.

## 1.9 Application setting

It is quite often that another web application may be required to access into the Ultrasearch application. One such application is Frontline a customer management application which is illustrated below.

To integrate the Ultrasearch into another application such as Frontline [a Lagan software application] such that it can be invoked directly from the Information menu item, the file `frontlineapplication.properties` within the `frontline/config` directory needs to be edited.

Change the line below:

```
application.appearance.information.default=http://apps.server.addr:7778/ultrasearch/query/search.jsp
```

where the http address is the address of the address where the ultrasearch instance is located.

## 2 SCREEN SETUPS

### 2.1 Primary Screen

Layout along lines of:

```
Topic:      xxxxxxxxxxxxxxxxxxxxxxxxxxxx No drop down - From TAG
Authority1  xxxxxxxxxxxxxxxxxxxxxxxxxxxx LOV[drop down] From TAG
Authority2  xxxxxxxxxxxxxxxxxxxxxxxxxxxx LOV[drop down] From TAG
Keywords    xxxxxxxxxxxxxxxxxxxxxxxxxxxx freetext
```

Authority is determined from the TAG DC.Coverage.Spatial.

Topic would be a search of the TAG DC.Subject.Category.

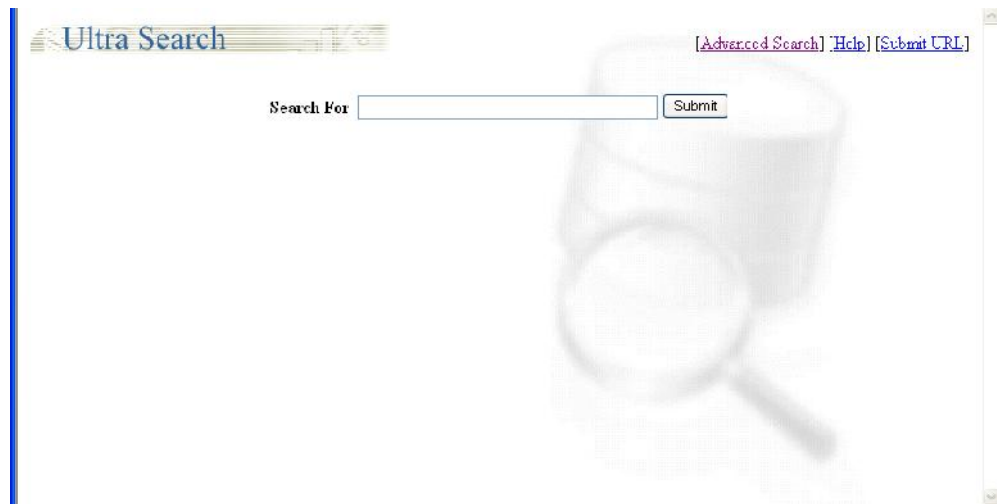
Keywords would be a freetext search of the pages themselves for content.

Current sample is shown below:

**Figure 1 - Default Ultrasearch screen**

### 2.2 Advanced Screen

Similar to current sample with ability to add more attributes to the search as required. Probably not change too much.



**Figure 2 - Default advanced Ultrasearch screen**

## **A. Appendix**

### **A.1 Sample Search\_JSP Code**

```
<%--
/*
-- Author : Macrotone Consulting Ltd
-- Date : 19/03/2005
-- Filename : _search.jsp
-- Version : 1.0
-- Description : Calls relevent JSPs to create a form to query the Ultrasearch DB and display the
results
-- Documented: in 'Base Installation Guide'

*/
--%><%!
    boolean l_force_advanced_mode = false;
    private static final boolean s_debug = false;
    boolean l_general_page = true;
%>
<%@page contentType="text/html; charset=UTF-8" %>
<%@page import = "oracle.ultrasearch.query.*" %>
<%@page import = "oracle.ultrasearch.query.app.*" %>
<%@page import = "java.io.UnsupportedEncodingException" %>
<%@page import = "java.lang.StringBuffer" %>
<%@page import = "java.util.*" %>
<%@page import = "java.math.BigDecimal" %>
<%@page import = "java.text.MessageFormat" %>
<%@page import = "java.text.SimpleDateFormat" %>
<%@page import = "java.sql.Statement" %>
<%@page import = "java.sql.Connection" %>
<%@page import = "java.sql.SQLException" %>
<%@page import = "java.sql.Types" %>
<%@page import = "java.sql.Array" %>
<%@page import = "java.sql.CallableStatement" %>
<%@page import = "javax.sql.DataSource" %>
<%@page import = "oracle.jdbc.pool.OracleDataSource" %>
<%@page import = "oracle.jdbc.pool.OracleConnectionPoolDataSource" %>
<%@page import = "oracle.sql.ArrayDescriptor" %>
<%@page import = "oracle.sql.ARRAY" %>
<%@page import = "oracle.sql.STRUCT" %>
<%@page import = "oracle.sql.Datum" %>
<jsp:useBean id="usearch_resBunHash" class="java.util.Hashtable"
scope="application"></jsp:useBean>
<jsp:useBean id="usearch_queryInstance" class="oracle.ultrasearch.query.QueryInstance"
scope="application"></jsp:useBean>
<jsp:useBean id="usearch_randomGenerator" class="java.util.Random"
scope="application"></jsp:useBean>
<%@page session = "false" %>
<%
    //
=====
=====
    // Include common functions
    //
=====
=====
%>
<%@ include file="common_global_code.jsp" %>

<%
// Set the style class for errors
```

```
String l_error_class = "colorSubheader";
try
{
    //
    =====
    // Include common initialization code
    //
    =====
    %>
    <%@ include file="common_init.jsp" %>
    <%@ include file="common_customize_instance.jsp" %>
    <%
        // Set the current request's locale

        // if container is servlet 2.0 or 2.1, then must parse Header for locale
        // l_request_locale = getRequestLocale(request);
        // if container is servlet 2.2 compliant, then use the following line instead
        l_request_locale = request.getLocale();

        // final fall-back
        if (l_request_locale == null)
            l_request_locale = Locale.US;

        l_request_collator = getBestCollator(l_request_locale);
        String l_lang_postfix = getLanguagePostfix(l_request_locale);
        String l_help_page;
        if (l_lang_postfix == null)
            // last resort if language is not supported
            l_help_page = "html/query_help.html";
        else
            l_help_page = "html/query_help_" + l_lang_postfix + ".html";
    %>
    <%@ include file="common_querytool_init.jsp" %>
    <%@ include file="querybox_application.jsp" %>

    <%
        List authorities = new ArrayList();

        authorities.add("Authority 1");
        authorities.add("Authority 2");
        authorities.add("Authority 3");

        int numAuths = authorities.size();

    %>

    <HTML>
    <HEAD>
        <TITLE><%=getMessage(l_resBun, "ORACLE_ULTRA_SEARCH")%></TITLE>
        <LINK href="/styles.css" rel="stylesheet" type="text/css">
    <script language="javascript" type="text/javascript">
    <%
        StringBuffer jsArray = new StringBuffer("var auth = new Array(");
        for(int i=0;i<authorities.size();i++)
        {
            jsArray.append("'" + (String)authorities.get(i) + "',");
        }
        jsArray.deleteCharAt(jsArray.length()-1);
        jsArray.append(");");

        out.print(jsArray.toString());
    %>
    </script>
    </HEAD>
    <BODY>
        <div class="colorSubheader">
            <h2><%=getMessage(l_resBun, "ORACLE_ULTRA_SEARCH")%></h2>
        </div>
    </BODY>
    </HTML>
    %>
    %>
```

```
%>

</script>
<SCRIPT src="dynamicForm.js" language="javascript" type="text/javascript"></SCRIPT>

</HEAD>
<BODY background="images/wsd.gif" onLoad="optionTestIt()">
<CENTER>
<TABLE width="%100" border="0" cellpadding=2 cellspacing=0>
<TR>
<TD width="1%" align="left">
<IMG src="images/ultra_mediumbanner.gif">
</TD>
<TD align="right" nowrap>
<font class="<%=l_qbox_font_main%>">
<%
/*
// Render the advanced or basic search link
if (showSearchLink == 1)
{
if ((advanced != null) && (advanced.equals("Basic")))
{
out.println("[ "+ getLink(l_qbox_advanced_link, l_qbox_text_advanced_search)+" ]");
} else {
out.println("[ "+ getLink(l_qbox_basic_link, l_qbox_text_basic_search)+" ]");
}
}
}
*/
// Render the help page link
String l_locale_string = "en_US";
if (l_request_locale != null) l_locale_string = l_request_locale.toString();
%>

</font>
</TD>
</TR>
<TR>
<TD colspan="2" width="100%">&nbsp;   </TD>
</TR>
<TR>
<TD align="center" colspan="2" width="100%">
<%
// -----
// Set the modes
// -----

l_show_results = true;

// -----
// The following are result related. Hence they should not have any
// equivalent in the query.jsp
// -----

l_hits_requested = "p_hitcount";

// title = data.getTitle();
dateshown = "1";
descriptionshown = "1";
sizeshown = "1";
scoreshown = "1";
authorshown = "1";
lhitsperpage = "10";
lpagelinks = "10";
lmaxpagenum = "1000";
```

```
add_more_attributes_button = getMessage(l_resBun, "ADD_MORE_ATTRIBUTES");

// -----
// The following are query related. Hence they must have an
// equivalent in the query.jsp
// -----

searchforlabel = l_text_search_for;
searchbutton = l_text_search;

l_show_querybox = true;

//
=====
// Include common query code
//
=====
%>
<%@ include file="common_query.jsp" %>
<%
//
=====
// Query Section
//
=====
%>
<%@ include file="_common_querybox.jsp" %>
<%
//
=====
// common result application code
//
=====
%>
<%@ include file="common_result.jsp" %>
<%
//
=====
// Page Link Section Begin
//
=====
%>
<%@ include file="common_pagelinks.jsp" %>
<%
//
=====
// Ending HTML
//
=====
%>
</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
<%
}
catch (Exception e)
{
```

```
e.printStackTrace(System.err);
out.println("<CENTER>");
out.println(" <TABLE border=\"0\" width=\"100%\">");
out.println(" <TR>");
out.println(" <TD class=\""+l_error_class+"\">");
out.println("     " + e);
out.println(" </TD>");
out.println(" </TR>");
out.println(" </TABLE>");
out.println("</CENTER>");
}
```

```
//
=====
// Functions used
//
=====
```

```
%>
<%!
public String getSearchPageLinkParams
(
    String advanced,
    String l_qbox_advanced_link,
    String l_qbox_basic_link,
    String firstHitpname,
    int startNum,
    String lastHitpname,
    int endNum,
    String linkvars
)
{
    StringBuffer b = new StringBuffer();
    b.append(firstHitpname);
    b.append("=");
    b.append(startNum);
    b.append("&");
    b.append(lastHitpname);
    b.append("=");
    b.append(endNum);
    b.append(linkvars);
    return b.toString();
}
```

```
public String getSearchPageLink
(
    String advanced,
    String l_qbox_advanced_link,
    String l_qbox_basic_link,
    HttpServletRequest request,
    String firstHitpname,
    int startNum,
    String lastHitpname,
    int endNum,
    String linkvars
)
{
    StringBuffer s = new StringBuffer();
    if (advanced!=null && advanced.equals("Advanced"))
    {
        s.append(l_qbox_advanced_link);
    }
    else
    {
        s.append(l_qbox_basic_link);
    }
}
```



```
    }  
    // don't use the request object at all  
    s.append("&").append(getSearchPageLinkParams  
        (advanced, l_qbox_advanced_link, l_qbox_basic_link, firstHitpname, startNum, lastHitpname,  
        endNum, linkvars));  
    return s.toString();  
    }  
%>
```

## **A.2 Sample Common\_Querybox.JSP Code**

```
<%--  
  
/*  
  
-- Author : Macrotone Consulting Ltd  
-- Date : 19/03/2004  
-- Filename : _common_querybox.jsp  
-- Version : 1.0  
-- Description : Creates a form to query the Ultrasearch DB  
-- Documented: in 'Base Installation Guide'  
  
*/  
  
--%><%  
  
    // Denuffify the main search string  
    query = denuffify(query);  
  
    // Prepare text strings  
    String l_qbox_text_any_language = getMessage(l_resBun, "ANY_LANGUAGE");  
    String l_qbox_text_attributes = getMessage(l_resBun, "ATTRIBUTE_SELECTION");  
    String l_qbox_text_from = getMessage(l_resBun, "FROM");  
    String l_qbox_text_language = getMessage(l_resBun, "LANGUAGE");  
    String l_qbox_text_and = getMessage(l_resBun, "AND");  
    String l_qbox_text_or = getMessage(l_resBun, "OR");  
    String l_qbox_text_add_more_attributes = add_more_attributes_button;  
    String l_specific_groups_checked = "";  
    String l_all_groups_checked = "";  
  
    // Utility strings to be used in the rendering process  
    String l_qbox_selected = "";  
    String l_qbox_oper_selected = "";  
    String l_qbox_attribute_value = "";  
    String l_hidden_attribute_id = "";  
    String l_qbox_querystring_value = "";  
    java.util.Locale l_qbox_lang = language;  
  
    if (allGroupsSelected.equals("specific"))  
    {  
        // The user opted to select specific data groups  
        l_all_groups_selected = false;  
        l_specific_groups_checked = "CHECKED";  
    }  
    else  
    {  
        l_all_groups_checked = "CHECKED";  
    }  
  
    // Handle the 'add more attributes' event.  
    // Increment l_attribute_rows if necessary in time for the hidden var  
    l_attribute_rows_int = 4; // min value is 4  
    if (l_attribute_rows != null)
```

```
{
  try
  {
    l_attribute_rows_int = Integer.parseInt(l_attribute_rows);
  }
  catch (Exception nex)
  {
    l_attribute_rows_int = 4;
  }
}
if (action!=null && action.equals(l_qbox_text_add_more_attributes))
{
  l_attribute_rows_int++;
  l_attribute_rows = Integer.toString(l_attribute_rows_int);
}

// Add the portlet ID (which may be an empty string) to various javascript function names.
l_formname = new String("usearchQbox"+l_portlet_id);
l_clear_funcname = new String(l_clear_funcname + l_portlet_id);
l_openwin_funcname = new String(l_openwin_funcname + l_portlet_id);

// Start drawing the form. Draw a bunch of hidden values that are required.
%>
<form onSubmit="cleanUp()" name="<%=l_formname%>" method="post"
action="./_search.jsp?usearch.p_mode=Advanced">
  <%= denuilify(l_qbox_hidden_form_fields) %>
  <input type="hidden" name="<%=l_usearch_nextview%>" value="Submit">
  <input type="hidden" name="<%=l_pname_firsthit%>" value="">
  <input type="hidden" name="<%=l_pname_lasthit%>" value="">
  <input type="hidden" name="<%=l_usearch_p_attrrows%>"
value="<%=l_attribute_rows%>">
  <input type="hidden" name="<%=l_pmode_paramname%>" value="<%=advanced%>">
<%
// If the search box is to be shown, draw it. Search portlets can hide the search box.
if (l_show_querybox)
{
  %>
  <TABLE border="0">

<%
// =====
// (L1) Everything below here is only when we're in advanced mode
// =====

  advanced= new String("Advanced");

  if ((advanced != null) && (advanced.equals("Advanced")))
  {
    //
    =====
    // (L2) Data Group Selection
    //
    =====
    int groupChoicesSize = 0;
    if (l_hide_datagroup_section.equals("checked"))
    {
      l_datagroup_section_hidden = true;
    }
    else
    {
      Collection l_col = l_instmd.getGroups();
      int l_col_size = 0;
      Iterator l_iter = null;
      oracle.ultrasearch.query.Group[] groupChoices = null;
```



```
        l_datagroup_checked = "";
        if (selectedGroups != null && selectedGroups.length > 0)
        {
            int j;
            for (j = 0; j < selectedGroups.length; j++)
            {
                if (groupId.equals(selectedGroups[j]))
                {
                    l_datagroup_checked = "CHECKED";
                }
            }
        }
    }
}
%>
        <INPUT TYPE="checkbox" NAME="<%=l_usearch_p_groups%>"
value="<%=groupId%>" <%=l_datagroup_checked%>"><%=htmlEncode(groupName)%>
<%
        if (ll_groupCount%3 == 0)
        {
%>
            </font>
            </td>
        </tr>
        <tr>
            <td>
                &nbsp;
            </td>
            <td align="left" nowrap>
                <font CLASS="<%=l_qbox_font_main%>">
<%
            }
        }
%>
            </font>
            </td>
        </tr>
<%
    }
}
}

//
=====
// (L2) Draw multiple rows for attribute conditions
//
=====
if (l_hide_attribute_section.equals("checked"))
{
    l_attribute_section_hidden = true;
}

if (!l_attribute_section_hidden && advanced != null && advanced.equals("Advanced"))
{
    operator = denulify(operator);
    String l_any_operator_checked = "";
    String l_all_operator_checked = "";
    if (operator.equals("any"))
    {
        l_any_operator_checked = "CHECKED";
    }
    else if (operator.equals("all") || operator.equals(""))
    {
        l_all_operator_checked = "CHECKED";
    }
}
%>
```

```
<tr><td>
<%
    String[] empty1 = { "", "", "", "" };
    String[] empty2 = { "-1", "-1", "-1", "-1" };
    String[] p_atnames = request.getParameterValues("usearch_p_atname") == null ?
empty1 : request.getParameterValues("usearch_p_atname");
    String[] p_ats = request.getParameterValues("usearch_p_at") == null ? empty2 :
request.getParameterValues("usearch_p_at");
    /*
        for (int i=0;i<4;i++)
        {
            out.println("pats: "+p_ats[i]);
            out.println("patnames: "+p_atnames[i]);
        }
    */
    String remTopic = request.getParameter("remTopic") == null ? "" :
request.getParameter("remTopic");
    String quoted = request.getParameter("keyQuoted") == null ? "" :
request.getParameter("keyQuoted");
%>
    <table border="0">
        <input type="hidden" name="remTopic" value="<%=remTopic%>">
        <input type="hidden" name="remKey"
value="<%=request.getParameter("remKey")%>">
        <input type="hidden" name="keyQuoted" value="<%=quoted%>">
        <input type="hidden" name="usearch_p_atname" value="<%=p_atnames[0]%>">
        <input type="hidden" name="usearch_p_op" value="contains">
        <input type="hidden" name="usearch_p_at" value="<%=p_ats[0]%>">
    <tr><td>
        <% String[] vals = request.getParameterValues("dusearch_p_val");
        //String pval = request.getParameter("remTopic") == null ? "" :
request.getParameter("remTopic");
        if(!remTopic.equals(""))
            {remTopic = "&quot;" + remTopic + "&quot;";}
    %>
        Topic:&nbsp;
        <input type="text" name="dusearch_p_val" size="50"
value="<%=remTopic%>">
        <input type="hidden" name="usearch_p_val" value="<%=remTopic%>">
    </td></tr>
    <input type="hidden" name="usearch_p_main_operator" value="all">
    <tr><td><table border="0">
        <input type="hidden" name="usearch_p_atname" value="Authority(String)">
        <input type="hidden" name="usearch_p_op" value="contains">
        <input type="hidden" name="usearch_p_at" value="142">
    <tr><td>
    <%String[] auths = request.getParameterValues("usearch_p_val");
    String authSearch;
    if(auths != null)
    {
        authSearch = auths[1];
    }else
    {
        authSearch = "";
    }
    %>
        Authority 1:&nbsp;
        <input type="hidden" name="usearch_p_val" value="<%=authSearch%>">
        <select name="dusearch_p_val" onChange="updateList2()">
    <%
    String all = "";
    String disable = "";
    if (request.getParameter("allselected")!=null)
    {
        if(request.getParameter("allselected").equals("true"))
```

```
        {
            all = "selected";
            disable = "disabled";
        }
    }else{disable = "disabled";}
%>
    <option value="All" <%=all%>> All </option>
<%
    for(int i=0;i<authorities.size();i++)
    {
        String selected = "";
        if(vals!=null)
        {
            if (all.equals("selected"))
            {
                selected = "";
            }else
            if(vals[1].equalsIgnoreCase((String)authorities.get(i)))
            {selected = "selected";}
        }
    }
    out.print("<option value='"+(String)authorities.get(i)+"' "+selected+">
    "+(String)authorities.get(i)+" </option>");
}
%>
    </select>
    <input type="hidden" name="allselected">
    </td><td>
    <input type="hidden" name="usearch_p_op" value="equals">
    <input type="hidden" name="usearch_p_at" value="-1">
    </td><td>
    Authority 2:&nbsp;
    <select name="dusearch_p_val" <%=disable%>>
    <option value=""> -- Choose -- </option>
<%
    for(int i=0;i<authorities.size();i++)
    {
        String selected = "";
        if(vals!=null)
        {
            if (all.equals("selected"))
            {
                selected = "";
            }else
            if(vals[2].equalsIgnoreCase((String)authorities.get(i)))
            {selected = "selected";}
        }
    }
    out.print("<option value='"+(String)authorities.get(i)+"' "+selected+">
    "+(String)authorities.get(i)+" </option>\n");
}
%>
    </select>
    </td>
    </td>
</tr></table>
</tr>
<tr><td>
<%
    String val = request.getParameter("remKey") == null ? "" :
request.getParameter("remKey");
    if (!quoted.equals(""))
    {
        val = "&quot;"+val+"&quot;";
    }
%>
%>
```

```
        Keywords:&nbsp;
        <input type="text" name="p_mainsearch" size="45" value="<%=val%>">
    </td></tr>
</tr><tr><td><hr></td></tr></tr>
<tr align="RIGHT">
    <td>
    <input type="submit" name="usearch.p_action" value="Submit">
    <input type="button" value="Reset" onclick="clearForm();">
    </td>
</tr>
<tr>
    <td>
<%
/*
Enumeration paramNames = request.getParameterNames();
while (paramNames.hasMoreElements())
{
    String name = (String) paramNames.nextElement();
    String[] values = request.getParameterValues(name);
    out.println(" <font color=\"red\">" + name + ": </font>");
    for (int i = 0; i < values.length; i++)
    {
        out.println(" " + values[i]);
    }
    out.println("<br>");
}
*/
%>
</td>
</tr>
</table>
<input type="hidden" name="usearch_p_lang" value="">
<input type="hidden" name="usearch_p_op" value="equals">
<input type="hidden" name="usearch_p_at" value="-1">
<input type="hidden" name="dusearch_p_val" value="">
<input type="hidden" name="usearch_p_atname" value="">
<%
}
}
%>
</td></tr>
</table>
<%
}
}
// The query box is not shown at all. Hence, we must
// write hidden elements to support the results-only portlet
if (!I_show_querybox)
{
    // Print the main search string
    out.println("<input type=\"hidden\" name=\"p_mainsearch\" value=\""+query+"\">");

    // Print the language
    if (language != null)
    {
        out.println("<input type=\"hidden\" name=\""+I_usearch_p_lang+"\"
value=\""+language+"\">");
    }
    else
    {
        out.println("<input type=\"hidden\" name=\""+I_usearch_p_lang+"\" value=\"\">");
    }
}
}
```

```
// The query box is hidden or datagroup section is explicitly hidden
// Therefore, we should draw the hidden data group elements
if (!l_show_querybox || l_datagroup_section_hidden)
{
    if (l_datagroup_section_hidden)
    {
        if (l_preselect_data_groups.equals("checked"))
        {
            // Override group selection mode to specific
            out.println("<input type=\"hidden\" name=\"" + l_usearch_p_allgroups + "\"
value=\"specific\">");
        }
        else
        {
            // Override group selection mode to all
            out.println("<input type=\"hidden\" name=\"" + l_usearch_p_allgroups + "\" value=\"all\">");
        }
    }
    else
    {
        // Print group selection mode (all/specific) according to request value
        out.println("<input type=\"hidden\" name=\"" + l_usearch_p_allgroups + "\"
value=\"" + allGroupsSelected + "\">");
    }
}

// Print out all selected groups
if (selectedGroups != null && selectedGroups.length > 0)
{
    int j;
    for (j = 0; j < selectedGroups.length; j++)
    {
        String groupId = selectedGroups[j];
        if (groupId != null && !groupId.equals(""))
        {
            out.println("<input type=\"hidden\" name=\"" + l_usearch_p_groups + "\"
value=\"" + groupId + "\">");
        }
    }
}

// The query box is hidden or attribute section is explicitly hidden.
// Therefore, we should draw the hidden attribute related elements
if (!l_show_querybox || l_attribute_section_hidden)
{
    // Print the main operator
    out.println("<input type=\"hidden\" name=\"" + l_usearch_p_main_operator + "\"
type=\"" + operator + "\">");

    // List out multiple rows of attribute/operator/attribute constraint
    if (arow != null)
    {
        for (int i = 0; i < arow.length; i++)
        {
            if (arow != null && i < arow.length && arow[i] != null)
            {
                // Print attribute id
                if (arow[i].id != null && !(arow[i].id.equals("")))
                {
                    out.println("<input type=\"hidden\" name=\"" + l_usearch_p_at + "\"
value=\"" + arow[i].id + "\">");
                }
                else
                {

```



```

        out.println("<input type=\"hidden\" name=\""+l_usearch_p_at+"\" value=\"\">");
    }

    // Print operator
    if (arow[i].operator != null && !(arow[i].operator.equals("")))
    {
        out.println("<input type=\"hidden\" name=\""+l_usearch_p_op+"\"
value=\""+arow[i].operator+"\">");
    }
    else
    {
        out.println("<input type=\"hidden\" name=\""+l_usearch_p_op+"\" value=\"\">");
    }

    // Print attribute constraint
    if (arow[i].query != null && !(arow[i].query.equals("")))
    {
        out.println("<input type=\"hidden\" name=\""+l_usearch_p_val+"\"
value=\""+arow[i].query+"\">");
    }
    else
    {
        out.println("<input type=\"hidden\" name=\""+l_usearch_p_val+"\" value=\"\">");
    }
    }
}
}
}
}
}
%>

</form>
<%
if (advanced!=null && advanced.equals("Advanced"))
{
    // =====
    // Javascript for handling attribute changes
    // =====
    // String l_server_name = portletRequest.getPageURL();
    // l_server_name = "oracle.com";
%>

<script language="JavaScript" type="text/javascript">
<!--
function <%=l_clear_funcname%>(formname)
{
    var myForm = eval("document." + formname);

    // Clear the main search box
    if (myForm.p_mainsearch.value)
    {
        myForm.p_mainsearch.value = "";
    }

    // First uncheck all data groups if necessary
    var myAllGroups = myForm.<%=l_usearch_p_allgroups%>;
    if (myAllGroups)
    {
        if (myAllGroups)
        {
            if (myAllGroups[0])
            {
                var myAllGroupsSelected = myAllGroups[0].checked;
                if (!myAllGroupsSelected)
                {

```

```
var myGroupChoices = myForm.<%=l_usearch_p_groups%>;
if (myGroupChoices)
{
  var numGroupChoices = myGroupChoices.length;
  for (var i=0; i<numGroupChoices; i++)
  {
    if (myGroupChoices[i])
    {
      myGroupChoices[i].checked = false;
    }
  }
}

// Set to Match all, not Match any
if (myForm.<%=l_usearch_p_main_operator%>)
{
  if (myForm.<%=l_usearch_p_main_operator%>[0])
  {
    myForm.<%=l_usearch_p_main_operator%>[0].checked = true;
  }
}

// Clear all attribute rows
if (myForm.<%=l_usearch_p_at%>)
{
  var numAttrRows = myForm.<%=l_usearch_p_at%>.length;
  for (var i=0; i<numAttrRows; i++)
  {
    if (myForm.<%=l_usearch_p_at%>[i])
    {
      myForm.<%=l_usearch_p_at%>[i].value = "-1";
    }
    if (myForm.<%=l_usearch_p_atname%> && myForm.<%=l_usearch_p_atname%>[i])
    {
      myForm.<%=l_usearch_p_atname%>[i].value = "";
    }
    if (myForm.<%=l_usearch_p_op%> && myForm.<%=l_usearch_p_op%>[i] &&
myForm.<%=l_usearch_p_op%>[i][0])
    {
      myForm.<%=l_usearch_p_op%>[i][0].selected = true;
    }
    if (myForm.<%=l_usearch_p_val%> && myForm.<%=l_usearch_p_val%>[i])
    {
      myForm.<%=l_usearch_p_val%>[i].value = "";
    }
  }
}

if (myForm.<%=l_usearch_p_lang%>)
{
  myForm.<%=l_usearch_p_lang%>.value = "";
}
}

function <%=l_openwin_funcname%>(datagroup_section_hidden, formname, elementidx,
lovmode, locale_string, showmode)
{
  var attrId = document.<%=l_formname%>.<%=l_usearch_p_at%>[elementidx].value;
```

```
var lurl =
"<%=l_lov_page%>?usearch.p_formname="+formname+"&<%=l_usearch_request_locale_string
%>="+locale_string+"&usearch.p_lov_aid="+attrId+"&usearch.p_showmode="+showmode+"&us
earch.p_lovmode="+lovmode+"&usearch.p_elementidx="+elementidx;

var myForm = eval("document." + formname);
var myAllGroups;
var myAllGroupsSelected;
var myAllGroupsSelectedValue;

if (datagroup_section_hidden)
{
  if (myForm.<%=l_usearch_p_allgroups%>)
  {
    myAllGroupsSelectedValue = myForm.<%=l_usearch_p_allgroups%>.value;
  }
  else
  {
    // If the all groups radio button doesn't exist, it's because
    // there are no groups defined. Behavior should be as if all
    // groups were selected.
    myAllGroupsSelectedValue = "all";
  }
}
else
{
  if (myForm.<%=l_usearch_p_allgroups%>)
  {
    myAllGroups = myForm.<%=l_usearch_p_allgroups%>;
    myAllGroupsSelected = myAllGroups[0].checked;
  }
  else
  {
    // If the all groups radio button doesn't exist, it's because
    // there are no groups defined. Behavior should be as if all
    // groups were selected.
    myAllGroupsSelected = true;
  }
}
if (myAllGroupsSelectedValue == "all" || myAllGroupsSelected == true)
{
  lurl += "&<%=l_usearch_p_allgroups%>=all";
}
else
{
  lurl += "&<%=l_usearch_p_allgroups%>=specific";
  var myGroupChoices;
  myGroupChoices = myForm.<%=l_usearch_p_groups%>;
  var numGroupChoices = 0;
  // If 0 or more groups exist
  if (myGroupChoices != "undefined")
  {
    // If exactly one group exists
    if (typeof myGroupChoices.value != "undefined")
    {
      // a single string
      numGroupChoices = 1;
      if (datagroup_section_hidden)
      {
        var myGroupChoiceValue = myGroupChoices.value;
        if (myGroupChoiceValue != null && !(myGroupChoiceValue == ""))
        {
          lurl += "&<%=l_usearch_p_groups%>=" + myGroupChoiceValue;
        }
      }
    }
  }
}
```

