Oracle® Ultra Search

Release Notes

Release 2 (9.0.2)

April 2002

Part No. A97355-01

This document summarizes the differences between Oracle Ultra Search in Oracle9*i* Application Server Release 2 (9.0.2) and its documented functionality.

To view the Ultra Search documentation:

- View the documentation with a Web browser from the Oracle documentation CD.
- View the documentation within an Ultra Search installation at `$ORACLE_HOME/ultrasearch/doc/help/toc.htm`.
- View the documentation through the Oracle Ultra Search Administration Tool by clicking the Help icon.

See Also: *Oracle9i Application Server Release Notes*

1 Ultra Search Installation

For installation documents within an Ultra Search installation, see `$ORACLE_HOME/ultrasearch/doc/help/install.htm`.

The Ultra Search middle tier is compliant with Oracle J2EE container (OC4J). Follow the instructions at `$ORACLE_HOME/ultrasearch/doc/help/install_midtier.htm` to configure the Ultra Search middle tier with OC4J.

Ultra Search requires you to have either JRE or JDK on the database host where you install the Ultra Search server component. By default, JDK 1.3.1 is installed by Oracle Universal Installer (OUI) during database installation under directory `$ORACLE_HOME/jdk`. If you use a different JDK, either create a soft link, or copy the files from the location where you install JDK

to $ORACLE_HOME/jdk. Ultra Search is certified with JDK 1.3.1 in this release.

2 Configuration Issues and Workarounds

This section describes Ultra Search configuration issues and their workarounds for Ultra Search.

2.1 Setting up Ultra Search Sample Query Applications

In addition to the configuration steps described in $ORACLE_HOME/ultrasearch/doc/help/install_midtier.htm, you must also follow these steps in order for Ultra Search sample query application to function correctly.

To configure Ultra Search sample query applications and sample search portlet, edit the OC4J data-sources.xml file. For editing data-sources.xml, see $ORACLE_HOME/ultrasearch/doc/help/install_midtier.htm.

2.1.1 9.0.1 Query API Sample - Deprecated in 9.0.2

To configure gsearch.jsp, you must manually edit the file and change the variable setting for 'username' and 'password', and then edit ultrasearch.properties to change the connection string. The file gsearch.jsp is under the $ORACLE_HOME/ultrasearch/sample/query/9i directory.

For editing the ultrasearch.properties, see $ORACLE_HOME/ultrasearch/doc/help/install_midtier.htm.

For detailed information, see: $ORACLE_HOME/ultrasearch/sample/query/9i/README.html

2.2 Running the Crawler

The Ultra Search Crawler is a Java process that runs on the server tier when launched. Therefore, Ultra Search requires you to have either JRE or JDK installed on the database host where you install the Ultra Search server component.

See Also: "Ultra Search Installation"

2.2.1 Stopping a Schedule

Stopping a schedule while it is running shuts down the crawler. Any documents that have not been crawled or indexed are processed when the crawler is restarted.

2.2.2 Setting the JOB_QUEUE_PROCESSES parameter

Oracle Ultra Search schedule launching uses the `DBMS_JOB` package. Therefore, the Oracle Ultra Search DBA must make sure that there is least one SNP process running. In other words, the initialization parameter file for the Oracle Ultra Search database instance should contain a line that specifies the `JOB_QUEUE_PROCESSES` parameter to be at least 1.

2.2.3 Verifying Crawler Progress

After you have configured the Oracle Ultra Search system, you can launch a crawler immediately from the Schedule screen.

View the Oracle Ultra Search crawler status by checking its state in the Oracle Ultra Search Administration Tool page.

Click the "Schedules" tab, and you see a table that lists all schedules and their state.

When a schedule is launching, it is in the "LAUNCHING" state. During the launching state, URLs to be crawled are copied into a queue. The amount of time this process takes depends on how many URLs there are to copy. In cases such as maintenance crawling of the Primary Schedule, there can be be millions of URLs to copy, with the schedule staying in the "LAUNCHING" state for a very long time.

When a schedule has completed launching and the crawler has begun fetching pages, the state changes to "EXECUTING".

2.3 Globalization Support

Oracle Ultra Search, like any other application that uses the database to store content, requires that the character set of the database be able to support the character set used at the application level. For example, if the application language is in Japanese, and if the database character set is SF7ASCII, then any data that the application attempts to store is corrupt, because the SF7ASCII character set does not support Asian languages.

The *Oracle9i Globalization Support Guide* provides detailed information on database character sets.

Note: Universal Character Sets, such as UTF8, attempt to support all languages of the world, including, but not limited to, Asian, European, and Middle Eastern languages.

3 Performance Tuning

3.1 Crawler Performance Tuning

For Crawler performance tuning, see Step 5 in Configuring the Oracle Server for Ultra Search at `$ORACLE_HOME/ultrasearch/doc/help/configure_server.htm`.

3.2 Query Performance Tuning

This section contains suggestions on how to improve the response time of the Ultra Search query.

3.2.1 Tune the DB_CACHE_SIZE Parameter

The database buffer cache keeps frequently accessed data read from datafiles, and efficient usage of the buffer cache can improve Ultra Search query performance. The cache size is controlled by the `DB_CACHE_SIZE` initialization parameter.

For more information on how to tune this parameter, see *Oracle9i Database Tuning Performance Guide and Reference*.

3.2.2 Optimize the Index

Optimize the Ultra Search index after the crawler has made substantial updates. This can be done by scheduling index optimization on a regular basis. Make sure index optimization is scheduled during off-peak hours, because query performance is slow during an index optimization schedule.

For information on index optimization schedules, see the Ultra Search online documentation about the Schedules Page (`$ORACLE_HOME/ultrasearch/doc/help/a_schedules.htm`).

3.2.3 Optimize the Index Based on Tokens

Optimize the Ultra Search index by basing it on frequently searched tokens. Queries can be logged by turning on query statistics collection in the Administration tool. The frequently searched tokens then can be passed to `CTX_DDL.OPTIMIZE_INDEX` in token mode. The Ultra Search index name is `WK$DOC_PATH_IDX`.

For more information on `OPTIMIZE_INDEX`, see *Oracle Text Reference*.

3.2.4 Simplify Query Expansion

The search response time is directly influenced by the Text query string used. Although Ultra Search provides a default mechanism to expand user input into a Text query, simpler expansions can greatly reduce search time.

For information on customizing query expansion, see the Ultra Search online documentation about Customizing the Query Syntax Expansion (`$ORACLE_HOME/qsyntax.htm`) and the Javadoc for the `oracle.ultrasearch.query.Query` interface.

4 Understanding Web Data Sources

This section explains in detail how Web data sources work. You should understand this section well before proceeding with crawling.

4.1 Introduction to Web Sources

Web sources are different from other data sources in the following:

1. There is exactly one default Web source that represents all of the Web pages collected. This default Web source is implicitly assigned to the default schedule.
2. There can be many user-defined Web sources where each is a "partition" or subset of the default Web source. Each user-defined Web source can represent only one Web site.

Take the following example:

- Default Web Source = www.foo1.com, www.foo2.com, www.foo3.com
- No user-defined Web sources

When the default schedule is launched, all of the URLs belonging to hosts www.foo1.com, www.foo2.com, and www.foo3 are collected and crawled.

Subsequently, if a user-defined Web source "foo2" is defined for "www.foo2.com", then URLs under www.foo2.com no longer belong to the default Web source. They now belong to the foo2 user-defined Web source. This means that when the default schedule is launched, foo2 URLs will not be crawled. Instead, a separate schedule needs to be created to crawl the foo2 data source.

When a user-defined Web source is dropped, the URLs of the dropped source are reassigned to the default Web source. They are not deleted from the system. So, if the "foo2" data source in the previous example is dropped,

then all of the foo2 URLs are re-assigned to the default Web source. They are then crawled whenever the default schedule is launched.

If you need to entirely eliminate all URLs that belong to a specific host, the only way to remove them from the system is to directly issue SQL statements in a SQL*Plus session. For example:

```
EXEC WK_ADM.USE_INSTANCE('<instance_name>');
DELETE FROM WK$URL WHERE URL LIKE 'http://www.foo2.com%';
COMMIT;
```

4.2 The Default Web Source

The default Web source is a collection of URLs that are discovered from a set of seed URLs. The set is bounded by one or more inclusion patterns and zero or more exclusion patterns. Specifically, the default Web source is defined by the following:

1. URL seeds; for example, "http://www.foo1.com/ http://www.foo2.com/ http://www.foo3.com/"
2. Inclusion patterns to define the boundary of the data source; for example, "foo1.com", "foo2.com", "foo3.com"
3. Optionally, exclusion patterns to define the exceptions within the inclusion set
4. Optionally, the crawling depth
5. Optionally, additional mime types like PDF documents

4.3 Maintenance Crawling

Maintenance crawling means that a page is not processed if it has not been changed since it was last crawled. To determine whether a page has changed, the crawler checks the Last Modified time stamp and the page checksum value. The checksum is based on the contents of the page. If the page has changed, then it is parsed again for new links and indexed.

For example, when launching a schedule that has the user-defined foo2 Web source associated with it, the crawler runs through all of foo2's URLs and possibly finds only a few URLs with changed pages. From those changed pages, the crawler might discover new URLs. A significant amount of time is saved because the crawler needed to process only URLs that have changed.

Note: If you want to process all pages again regardless of whether or not they have changed, then update the "Crawler Recrawl Policy" for each schedule by clicking the Edit icon in the Administration Tool. Select "Process all documents". You might want to do this is if you forgot to include a certain document type (for example, Microsoft Word) from the start and now would like the crawler to reprocess all pages to look for any links to Microsoft Word documents.

4.4 Seed URLs

A set of seed URLs can be defined as the starting points for the crawler to discover URLs. As URLs are discovered, they are assigned to a user-defined Web source, if the host name matches. Otherwise, they are assigned to the default Web source. A seed URL is useful only when it matches one of the inclusion patterns defined and is not excluded. There is no limit on the number of seeds that can be defined.

Here are two examples:

- Case #1
  - The default Web source has seed = "http://www.foo1.com/", inclusion = "www.foo1.com".
  - No user-defined Web sources.
  - Running the default schedule collects all URLs at host www.foo1.com and assigns them to the default Web source.
  - Running the default schedule again performs maintenance crawling on www.foo1.com.
  - Add a new seed "http://www.foo2.com/", and add "www.foo2.com" to the inclusion domain list.
  - Running the default schedule collects all URLs at host www.foo2.com and does maintenance crawling on already existing foo1 URLs.
- Case #2
  - The default Web source has seed = "http://www.foo1.com/", inclusion = "www.foo1.com".
  - A user-defined Web source "foo1" exists for Web site "http://www.foo1.com".

- Running the default schedule collects all URLs at host www.foo1.com, but they are now assigned to the foo1 user-defined Web source.
- Note that the URLs collected by the default schedule are controlled by the inclusion / exclusion patterns.
- Running the default schedule will not perform maintenance crawling on www.foo1.com URLs, because those URLs now belong to the foo1 user-defined Web source.
- To perform maintenance crawling on the www.foo1.com, you must define a new schedule and assign the foo1 Web source to it.

4.5 Defining the Default Web Source

Careful planning is needed when defining the default Web source. Specifically, the user must determine whether to build it using a top-down or bottom-up approach. **It is strongly recommended that the bottom-up approach be used for ease of management.**

4.5.1 Bottom-Up Approach

Web sites are incrementally added to the default Web source. The idea is to crawl all known important Web sites before allowing the crawler to discover unknown Web sites. The following sequence describes the process of incrementally adding Web sites:

1. Add the "www.foo1.com" inclusion pattern and "http:/ / www.foo1.com" as a seed to start.
2. Run the default schedule to completion.
3. Add a second inclusion pattern and seed.
4. Run the default schedule to completion again.
5. Note that the second time the default schedule is launched, foo1 URLs area crawled again for maintenance crawling. To avoid that, define "www.foo1.com" as a Web source right from the beginning or right before the second time the default schedule is launched.
6. Repeat steps 3 and 4 until all known important Web sites are crawled.
7. After all of the important Web sites are crawled, you can consider relaxing the inclusion rules to pick up the rest of the Web sites.

4.5.2 Top-Down Approach

The inclusion patterns are defined to contain an unknown number of Web sites; for example, the "oracle.com" inclusion pattern allows crawling of

URLs on any Web site within Oracle Corporation. The problems with this approach are as follows:

1. The number of URLs covered can be extremely large, and it might take a few weeks or months to finish the initial default schedule crawling.
2. There might be Web sites that, upon inspection, should not be crawled. But by the time they are discovered, it could be very tedious to remove all the URLs for that Web site from the crawler queue and URL table (which requires issuing SQL statements directly).
3. An unknown Web site might have millions of URLs that causes the crawler to "lock-in" to that Web site, so that no other Web sites are crawled or a long period of time.
4. The crawling depth required for individual Web sites might be different.

Note: The previous sequence incrementally crawled one Web site. You could incrementally crawl more than one Web site for each iteration.

4.6 URLs Crawled When a Schedule is Launched

Whenever a schedule is launched, the list of URLs to be crawled are enqueued into a queue before the crawler is run. The enqueuing of URLs differs depending on whether it is the default schedule or a schedule containing user-defined Web sources:

4.6.1 Default Schedule

The default schedule implicitly crawls the default Web source. Therefore, any new seed URLs (that were added after the last default schedule ran) are enqueued first. Then, all other URLs that are assigned to the default Web source are enqueued for maintenance crawling. URLs that belong to hosts for which user-defined Web sources exist are not enqueued. For example, if foo2 was defined as a Web source for host www.foo2.com, then URLs that begin with www.foo2.com are not enqueued.

4.6.2 Schedules Containing User-Defined Web Sources

Schedules that have user-defined Web sources associated URLs belonging to all associated user-defined Web sources are enqueued. For example, if you associate the foo2 Web source, then all URLs that belong to host www.foo2.com are enqueued.

Note: For both the default schedules and for schedules containing user-defined Web sources, if both www.foo1.com and www.foo2.com are specified as inclusion patterns, then URLs belonging to host www.foo2.com will be crawled even though those URLs were not enqueued at start up time. This happens if any page at www.foo1.com has links to pages at www.foo2.com. This is why you must carefully redefine the inclusion and exclusion domains each time you run a schedule.

5 Known Bugs

5.1 Bug 2219746 - Environment Not Getting Passed To Child Process Launched From JAVAVM On NT

The path of the JDBC OCI driver is not passed to the Ultra Search crawler JVM. As a result, the crawler cannot communicate with the database, and none of the crawled documents are indexed. By default, Ultra Search uses JDBC thin driver; therefore, no additional steps are needed for Ultra Search to work.

If you choose to use JDBC OCI driver for crawling, then Ultra Search requires `%ORACLE_HOME%/bin` to be part of NT system environment variable `%PATH%`. During Oracle installation, OUI makes the correct configuration to `%PATH%`. You must reboot the computer right after installation for this configuration to take affect.

5.2 Bug 2166510 - NLS: Crawler Shutdown Unexpected If Log Directory Name Contains MBCS

The crawler is unable to handle log directory path with multibyte characters.

Avoid specifying log directories that have Chinese, Japanese, or Korean characters.

5.3 Bug 2166662 - NLS: Can't Find Files With MBCS Name

File data source crawling cannot crawl directories or files with multibyte characters; for example, Chinese or Japanese.

Avoid naming the file in Chinese, Japanese, or Korean or putting them under such directory.

5.4 Bug 2186745 - File Data Source Can Not Crawl Directory Or File Name With HTML Reserved Symbol

The crawler is unable to pick up files or directories whose name contains HTML reserved symbols, like '<' or '>', when doing file data source crawling.

Rename the file or directory that is using such symbol.

5.5 Bug 2265110 - Unable To Stop Crawling When Crawler Is Enqueuing URL From A Crawler Agent

Stop crawling does not stop the crawler, even though the schedule status shows that the crawler has stopped. This happens when the crawler agent is used and the crawler is in the process of enqueuing URLs fetched from the agent. The crawler stops only after enqueuing is finished.

Currently, there is no way to stop the enqueuing other than manually killing the crawler process.

5.6 Bug 2114417 - Unwanted/Invalid Syntax Displayed For Query String

When query statistics collection is enabled, the query statistics pages (Daily summary of query statistics, Top 50 queries, Top 50 ineffective queries, Top 50 failed queries) may show text query strings like '(((WKA2X &({abc}))WITHIN S2))*2,({abc}))'. This behavior is due to 9.0.2 Java API expanding the user's query ('abc' in this case) before it is sent to the database.

In most cases, the user's query can be deciphered from the text query.

5.7 Bug 2262593 - Steps For Defining A Crawler Agent Type Need Improvement

When a user tries to create a new data source type, in Step 1, with regard to "Agent java class name" entering information about crawler agent, the user should enter just the class name without `.class` extension.

5.8 Bug 2218987 - Display URL For Items Of Type Page Link Are Not Unique

If a Portal item or page is of type page link, then its display URL will be a duplicate of the actual item / page being linked to. It is possible that multiple items link to one item (likewise for pages). Given a set of items all

of which have the same display URL, Ultra Search is only able to index exactly one of them.

Which Ultra Search indexes depend on which item is reached first by the crawler. Ensure that no page links or item links are created (pages or items are created in Portal that point to other existing pages or items in that same Portal).

5.9 Bug 2244234 - Translations Of Items Have Same Display URL In Portal XML

In Portal, the URLs for translations of an item have the same display URL as that of the base language item. Portal users can view different translations because when users log in to Portal, the language is established as part of the browser session. However, this language negotiation process only works with browsers operated by human users. Therefore, the Ultra Search crawler receives the same display URL for the translated items. This violates the stated requirement that all display URLs presented to Ultra Search be unique. The implication is that Ultra Search cannot crawl translations of an item.

As in bug 2218987, with multiple translations, only one of the translation items or the base item itself is indexed by the crawler. The rest are rejected by the crawler because of the duplicate display URLs.

5.10 Bug 2244239 - Some Attributes In Portal XML Are Not Encased In Translations Section

If there are translations for an item or page, then some attributes of that item / page cannot be correctly transmitted to the Ultra Search crawler. As a result, attribute queries may not work correctly for translated items.

5.11 Bug 2244254 - Portal XML Should Not Reveal Item Types Of BASETYPE=NONE

Ultra Search crawler can process specific Portal item types. However, Portal item type of "none" does not have display URLs. As a result, they are not revealed to Ultra Search. Because anything that does not have a display URL cannot be represented in the search application search results list in such a way that the user can click on it to view the item.

5.12 Bug 2097381 - Portal Users Should Embed Ultra Search Portlets That Are Hosted On The Same Host

Portal users should embed Ultra Search portlets that are hosted on the same host as the Oracle9*i*AS Portal server. If Oracle Portal is installed on host A, then Ultra Search is also installed on host A. Hence, the Ultra Search provider is also hosted as a Web application on host A.

It is possible that the Ultra Search provider running on host A could be registered with a second Oracle Portal instance running on host B. However, if the Ultra Search portlet hosted on A is embedded within pages created in Portal B, then the pop-up list-of-values will not work correctly. This is because of an security bug inherent in Javascript.