

Database Configurations

Table of Contents

Overview.....	2
Configuration Information.....	2
Database Jobs.....	2
Configuration of Database Jobs (General).....	2
mdClassifieds/mdRealEstate Configuration.....	3
Ad Retention.....	3
Database Jobs (mdClassifieds/mdRealEstate).....	3
PSearch Configuration.....	5
Database Jobs (PSearch).....	5
Database Email Configuration.....	5
Database Reference Information.....	5
Init.ora Parameters.....	5
Version/Patches.....	6
OS Kernel Parameters (Unix/Linux).....	6
Support Information.....	6

Overview

This manual provides information pertaining to database configurations, patches, and parameters.

Configuration Information

This section covers general database configuration, followed by some product-specific configuration information. Only those products that require special configuration are included.

Database Jobs

The SYS.DBMS_JOB package executes stored procedures that sync the Intermedia Text indexes and gather statistics for the MDW schemas. The frequency at which these jobs run can be configured to meet your business rules. This package also executes some product- (i.e. schema-) specific jobs (see the following product-specific sections below). Certain jobs related to products that you have not purchased (specifically those jobs that send email) are disabled by default. There is no need for these jobs to run unless you are running the products that utilize those jobs.

Configuration of Database Jobs (General)

(**Note:** The SQL statements in this section and those that follow can be used if you do not have access to any third-party or GUI tools that allow you to view/maintain DBMS jobs.)

The jobs that sync and optimize the Text indexes are owned by user CTXSYS. The sync jobs call the stored procedure CTXSYS.CTX_DDL.SYNC_INDEX(). Intermedia Text indexes need to be synced to process any inserts, updates, or deletes that have affected the indexed columns, since these DML activities aren't reflected in the indexes until the indexes are resynced. The default is to sync each index every 10 minutes. You can specify how often you want to sync each index by using the SYS.DBMS_JOB.INTERVAL(job, interval) procedure, where interval is expressed in days. The job argument is the id of the job. To obtain the job's id, and current job call (the "what" column) use the following query:

```
select job, what
from dba_jobs
where schema_user = 'CTXSYS';
```

Once you have the job's id, you can reset its interval. For example, to change job 14 to run every 30 minutes, execute this command:

```
exec sys.dbms_job.interval(14, 'sysdate + 30 / (24 * 60)');
commit;
```

The CTXSYS.CTX_DDL.OPTIMIZE_INDEX() procedure is used to maintain the efficiency of indexes. By default indexes are optimized weekly. You can change this interval using the same method previously given for configuring the sync index jobs.

In addition to the Intermedia Text index maintenance jobs, there is one job in each MDW schema that gathers statistics for that schema. By default these jobs run once weekly. The methods above for querying and modifying the CTXSYS jobs can be applied to these statistics-gathering jobs as well.

mdClassifieds/mdRealEstate Configuration

Ad Retention

Ads are archived based on the category they're in. Each category can be marked to have its ads archived or not. This setting can be updated via Template Manager, one of the GUI user applications. The default setting when a new category is created is *not* to archive the ads in that category. The stored procedure that archives ads takes as one of its parameters the number of days to keep ads online after they've expired (i.e. the end date has passed). All of these values are fully configurable via modification of the jobs that execute the stored procedures.

Database Jobs (mdClassifieds/mdRealEstate)

In mdClassifieds/mdRealEstate, there are a few jobs in the MDW_AD schema related to classified ad data. To get the information needed to change these jobs, run the following query:

```
select job, what
from dba_jobs
where schema_user = 'MDW_AD';
```

There is one materialized view used for ad features. The SYS.DBMS_REFRESH.REFRESH() procedure is used to refresh this view. The default is to refresh the view every 10 minutes. Use the SYS.DBMS_JOB.INTERVAL(job, interval) procedure to change this interval.

There are two jobs that, by default, run nightly. The first job archives ads and the second job expires ads. It is recommended that you not change the interval of these two jobs unless you change the times that the jobs run. To change the time at which a job executes, use the SYS.DBMS_JOB.NEXT_DATE(job, next_date) procedure. Use the query above to get the job number to change. For example, if job 21 normally runs at 1:00 AM and you want to have it run at 2:00 AM, execute the following (assuming the current date is 08/20/2002):

```
exec sys.dbms_job.next_date(21, to_date('08/21/2002 02:00',
'MM/DD/YYYY HH24:MI'));
commit;
```

You will also need to change the interval:

```
exec sys.dbms_job.interval(21, 'trunc(sysdate + 1) + 2/24');
commit;
```

Note how the interval was expressed in the last example. You can specify `sysdate + 1` as an interval for jobs that run once daily; however, the interval starts at the time the job completes. This means that if a job starts at 2:00 AM but takes 30 minutes to run, an interval of `sysdate + 1` will cause the job to begin at 2:30 AM the following day, and at 3:00 AM the day after. These cumulative changes would eventually cause the job to run much later than planned, which might adversely impact other production activities. Specifying the interval as `trunc(sysdate + 1) + 2/24` forces the job to start at 2:00 AM every day regardless of when it completes.

Ad archiving is handled by the MDW_AD.AD_MAINT.ARCHIVE_ADS(cutoff_date, sender, recipients) procedure. The default value for cutoff_date is `sysdate - 8`. Ads with end_dates before the cutoff_date will be archived. Setting the cutoff_date to `sysdate - 8` allows ads to be available online for a week after they've expired. This allows users to view last Sunday's ads in addition to the current day's ads, which is a fairly popular feature. The ad archiving process will email a message containing the number of ads archived. The email will be sent "from" the email address supplied in the sender argument. The recipient's argument is a comma-separated list of email addresses to receive the output of the ad archiving

process. Should an error occur, the email message will contain the text of the error. (See *Database Email Configuration* below, for instructions on configuring the stored procedure that sends the email.)

The procedure to expire ads, `MDW_AD.AD_MAINT.EXPIRE_ADS(sender, recipients)`, is similar to the procedure that archives ads. However, it takes only sender and recipients as arguments. This procedure sets the ad status to 'EXPIRED' for all ads whose `end_date` has passed. An email will be sent indicating the number of ads that were expired; if an error occurs, the email will contain the text of the error.

To change the arguments for these jobs use the `SYS.DBMS_JOB.WHAT(job, what)` procedure. For example, let's say you want to change the recipient's argument for the expiration job. First, run the following query to get the current job calls (the "what" column):

```
select job, what
from dba_jobs
where schema_user = 'MDW_AD';

JOB WHAT
-----
21 ad_maint.archive_ads(sysdate - 8, 'ad_archiver@mydomain.com',
'john@mydomain.com');
28 ad_maint.expire_ads('ad_expirer@mydomain.com',
'john@mydomain.com');
```

To change the recipient's argument to "bob@mydomain.com", do this:

```
exec sys.dbms_job.what(28,
'ad_maint.expire_ads(''ad_expirer@mydomain.com'',
''bob@mydomain.com'');');
commit;
```

Note that quotes within the "what" argument have to be passed as two single quotes.

It should be noted that the classified ad renderers (both static and dynamic) take a publication's time zone setting, also set through Template Manager, into account when retrieving ads. However, the ad expiration job does not.

Therefore, an allowance needs to be made for this when scheduling the time that the ad expiration job runs. In general the allowance should be the number of hours difference between the time zone where the database server resides and the time zone of the publication that is furthest behind the database server's time zone. For example, if the database server is in the Eastern Time zone, and there is a publication in the Pacific Time zone, the allowance is three hours and the ad expiration job should not run before 3:00 AM. Similarly, if the database server is the same time zone as all of the publications, or if the database server is in a time zone that is behind all of the time zones for the publications, the ad expiration job can run just after midnight.

PSearch Configuration

Database Jobs (PSearch)

If you are using the Persistent Search feature of mdClassifieds/mdRealEstate, there are two additional jobs that will need to be configured. Both of these jobs are in the MDW_REG schema. The first job, MDW_REG.PSEARCH_AUDITOR.EXPIRE_PSEARCH(sender, recipients), is used to expire stored searches whose expiration date has been reached. An email will be sent indicating the number of ads that were expired; if an error occurs, the email will contain the text of the error. (See *Database Email Configuration* below for instructions on configuring the stored procedure that sends the email.) A second job, MDW_REG.PSEARCH_AUDITOR.DELETE_PSEARCH_EMAIL_LOG(days_past, sender, recipients), is used to remove entries from the email log that are older than days_past old. The default is to keep 60 days worth of log entries. These jobs can be viewed/modified using the methods shown above, except substitute MDW_REG for MDW_AD:

```
select job, what
from dba_jobs
where schema_user = 'MDW_REG';
```

Database Email Configuration

(**Note:** Unless you are executing one of the above database jobs that send email, you will not need to perform this configuration.)

There are a few database jobs that email output to designated recipients. For these jobs (expire/archive classified ads, expire PSearch, delete PSearch email logs), there is a stored procedure that gets called to send the email notifications. This procedure, MDW_UTIL.SEND_EMAIL, needs to be configured to use a valid SMTP server to which it will connect. The following variables need to be set:

v_smtp_server – fully qualified name of the SMTP server to connect to
v_smtp_domain – domain of the SMTP server

Database Reference Information

Init.ora Parameters

Mandatory:

- DB_BLOCK_SIZE >= 8192
- JOB_QUEUE_PROCESSES >= 4
- JOB_QUEUE_INTERVAL = 60
- NLS_DATA_FORMAT = 'DD-MON-RR'
- COMPATIBLE = database version (e.g. "8.1.7")
- QUERY_REWRITE_ENABLED = TRUE
- QUERY_REWRITE_INTEGRITY = TRUSTED

(The last two parameters are required to allow the optimizer to use function-based indexes.)

Recommended:

- OPEN_CURSORS: >= 300
- PROCESSES: >= 200
- SORT_AREA_SIZE: >= 1 MB
- SORT_AREA_RETAINED_SIZE: same as SORT_AREA_SIZE
- JAVA_POOL_SIZE: >= 20MB
- CURSOR_SHARING = FORCE, if running at least version 8.1.7.3 of Oracle due to minimal use of bind variables in the application.
- event="32333 trace name context forever, level 10" to prevent materialized view refresh hangs. This is a known bug (1912308) that is fixed in version 9.0.2.

Version/Patches

Our software is tested to run on version 8.1.7 of Oracle Enterprise Edition; however, we recommend applying the latest patch set available for your platform.

OS Kernel Parameters (Unix/Linux)

- SEMMNS: 2 x PROCESSES parameter in init.ora
- SHMMAX: large enough to accommodate the size of the SGA.

Support Information

Morris Digital Works (MDW), a division of Morris Communications Co., provides tools, technologies, consulting and Web development services to Morris newspapers and external clients. MDW award-winning technologies include; world-class hosting facilities, robust content management software, high performance application tools, site enhancement tools and comprehensive classified and display classified technology. Founded in 1995, Morris Digital Works has over 100 employees with offices in Topeka, KS, Joplin, MO, New York, NY and headquartered in Augusta, GA. An additional 250 MCC employees also participate in our Internet business and report directly to newspapers, magazines, book publishing and other internal organizations.

If you are experiencing problems with any of Morris Digital Works products, please contact customer support at (706) 828-2955 ext. 2.