

## Changes in the Management of Cost-Based Optimizer Statistics in PeopleTools 8.48

by David Kurtz, Go-Faster Consultancy Ltd.

*A recent thread on Oracle-L<sup>1</sup> has led me to look at how Oracle has changed the way that PeopleTools 8.48 collects Oracle Cost-Based Optimizer statistics. It now uses DBMS\_STATS instead of the ANALYZE command. This has also caused me to reconsider some options for managing statistics for a PeopleSoft system.*

Application Engine programs can collect Cost-Based Optimizer statistics on specific tables by calling the `%UpdateStats(<table_name>[,high/low]);` PeopleCode macro. This uses one of two DDL models depending on whether the high or low option is specified. However, these DDL models only exist for Oracle and DB2/MVS. `%UpdateStats()` has no function on other platforms.

This was PeopleSoft's platform generic solution (before their takeover by Oracle, and before Dynamic Sampling was available in the Oracle database) to the very real problem that occurs when statistics on a working storage or reporting table, that is emptied, repopulated and used during a batch process, do not accurately represent the content of the table and hence cause the Optimizer to choose an

inappropriate execution plan.

PeopleSoft provided a method of refreshing the statistics during the process, and introduced new DDL models because each database platform would have its own command. However, this approach relies upon developers to add the `%UpdateStats()` macro for every occasion where data is changed sufficiently to require refreshing the statistics. Unfortunately, developers are not always best placed to make that decision. There are also plenty of places in delivered code where this macro could be usefully added.

However, remember that `%UpdateStats` implies a database commit because both the ANALYZE and DBMS\_STATS package commit the update of the statistics. Hence, `%UpdateStats` is not executed when called within a SELECT/FETCH loop in a restartable Application Engine program in order to prevent a commit being issued during that loop. A message is written to the log file.

```
11.30.18
... (FS_VATUPDFS.BB000.BB000-3) (SQL)
RECSTATS PS_VAT_UPD_BU_TAO LOW
/
11.30.18 UpdateStats ignored -
COMMIT required
```

Up to PeopleTools 8.47, PeopleSoft delivered two DDL models that used the ANALYZE command. The `%UpdateStats(,high)` ran a full compute of the table:

```
ANALYZE TABLE [TBNAME] COMPUTE
STATISTICS;
```

While `%UpdateStats(,low)` estimated statistics with the default sample size:

```
ANALYZE TABLE [TBNAME] ESTIMATE
STATISTICS;
```

<sup>1</sup> See <http://www.freelists.org/archives/oracle-l/04-2007/msg00959.html>

From PeopleTools 8.48, these DDL models now call the Oracle supplied PL/SQL package *DBMS\_STATS*. The high option estimates statistics using a 1% sample<sup>2</sup>.

```
DBMS_STATS.GATHER_TABLE_STATS
(ownname=> [DBNAME],
tablename=>[TBNAME],
estimate_percent=>1, method_opt=>
'FOR ALL COLUMNS SIZE
1', cascade=>TRUE);
```

While the low option estimates statistics with the sample size determined by the pseudo-variable.

*DBMS\_STATS.AUTO\_SAMPLE\_SIZE*

```
DBMS_STATS.GATHER TABLE STATS
(ownname=> [DBNAME],
tablename=>[TBNAME],
estimate_percent=>
dbms_stats.auto sample size,
method_opt=> 'FOR ALL INDEXED
COLUMNS SIZE 1', cascade=>TRUE);
```

So it would appear that PeopleSoft now follows the recommendations that Oracle have been making since version 8i of the database to use *DBMS\_STATS* instead of the *ANALYZE* command. This is certainly a step in the right direction. It also makes good sense to use the automatic sample size<sup>3</sup>.

However, there is a problem.

PeopleSoft have chosen to specify the *METHOD\_OPT* as *FOR ALL INDEXED COLUMNS SIZE 1*. If you have specified histograms on any of your columns, or generated them automatically with *DBMS\_STATS*, the PeopleSoft command will remove them from indexed columns and will leave

any histograms on unindexed columns unchanged, and potentially out of date.

The default in Oracle 9i is *FOR ALL COLUMNS SIZE 1*. This removes all histograms on all columns, although this is at least the same behaviour as the *ANALYZE* command.

In Oracle 10g *METHOD\_OPT* defaults to *FOR ALL COLUMNS SIZE AUTO*. The Oracle manual states that the database 'determines the columns to collect histograms based on data distribution and the workload of the columns'. However, Oracle will remove histograms if it judges that they are not necessary.

I would have no hesitation in recommending that value for *METHOD* the delivered DDL model should be changed to update histograms for *ALL COLUMNS* and not just indexed columns. I prefer to use *FOR ALL COLUMNS SIZE REPEAT* and to take control of which columns have histograms.

## Optimizer dynamic sampling

Optimizer dynamic sampling was introduced in Oracle 9.0.2 as a solution to the same problem. When a query is compiled Oracle can collect some Optimizer statistics based upon a small random sample of blocks for tables which do not have statistics and which meet certain other criteria depending upon the parameter *OPTIMIZER\_DYNAMIC\_SAMPLING*. In Oracle 10g, the default value for this

---

<sup>2</sup>I am sure this is a bug, and that PeopleSoft/Oracle intended that this should be a full compute with a 100% sample size.

<sup>3</sup>*ESTIMATE\_PERCENT* defaults to *DBMS\_STATS.AUTO\_SAMPLE\_SIZE* from Oracle 10g. Previously it was *NULL*, which caused a full compute.

parameter changed from 1 to 2 and so Dynamic Sampling applies to ALL unanalyzed tables.

Thus it should be possible to resolve the problem of incorrect statistics on a working storage table without explicitly collecting statistics during an Application Engine program, and therefore without needing a code change to add `%UpdateStats()`.

Instead, simply delete statistics from the table, and lock them so that a subsequent GATHER\_SCHEMA\_STATS will skip any locked tables. When a query references the table it will dynamically sample statistics and use them in determining the execution plan.

However, there is one more problem to overcome. Both the ANALYZE command and GATHER\_TABLE\_STATS will raise an error when run on a table with locked statistics.

```
SQL error. Stmt #: 1603 Error
Position: 0 Return: 20005 - ORA-
20005: object statistics are locked
(statttype = ALL) ...
```

If you lock statistics on a table, in order to use Dynamic Sampling on a table where `%UpdateStats()` is already used to update the statistics, the PeopleCode macro will raise an exception that will cause Application Engine programs to terminate with an error.

You could take the view that you will never gather statistics on any tables with `%UpdateStats`, but will always lock them without statistics and use Dynamic Sampling. In which case, it is

possible to disable the macro by setting the `Dbflags` bitfield in the Application Sever and Process Scheduler configuration files.

```
; DbFlags Bitfield
; Bit      Flag
; ---      ----
; 1        - Ignore metaSQL to
update database statistics(shared
with COBOL)
DbFlags=1
```

In practice, I think a finer degree of control is required.

A more flexible workaround is to encapsulate GATHER\_TABLE\_STATS in a procedure that handles the exception, and reference the procedure in the DDL model (it is not possible to put a PL/SQL block directly in the DDL model)<sup>4</sup>.

```
CREATE OR REPLACE PACKAGE wrapper AS
  PROCEDURE ps_stats (p_ownname
VARCHAR2, p_tabname VARCHAR2,
p_estpct NUMBER);
END wrapper;
/

CREATE OR REPLACE PACKAGE BODY
wrapper AS
  PROCEDURE ps_stats(p_ownname
VARCHAR2, p_tabname VARCHAR2,
p_estpct NUMBER) IS
  table_stats_locked EXCEPTION;
  PRAGMA
EXCEPTION_INIT(table_stats_locked,-
20005);
  BEGIN
    IF p_estpct = 0 THEN
      sys.dbms_stats.gather_table_stats
        (ownname=>p_ownname
        ,tabname=>p_tabname
        ,estimate_percent=>DBMS_STATS.AUTO_S
AMPLE_SIZE
        ,method_opt=>'FOR ALL COLUMNS
SIZE REPEAT'
        ,cascade=>TRUE);
    ELSE
      sys.dbms_stats.gather_table_stats
        (ownname=>p_ownname
        ,tabname=>p_tabname
        ,estimate_percent=>p_estpct
        ,method_opt=>'FOR ALL COLUMNS
SIZE REPEAT'
        ,cascade=>TRUE);
    END IF;
  EXCEPTION
    WHEN table_stats_locked THEN NULL;
  END ps_stats;
END wrapper;
/
```

<sup>4</sup> This package can be downloaded from <http://www.go-faster.co.uk/wrapper848.sql>

Thus it is possible for the DBA to choose whether to update statistics or use Dynamic Sampling for each table.

One more variant on this theme is to add the force=>TRUE option to GATHER\_TABLE\_STATS so that it will update statistics on locked objects. This is almost the reverse of the wrapper package. It will prevent statistics being refreshed by schema-wide jobs at inopportune times, but still let them be gathered by the application when appropriate. This could be done in the original DDL models without the need for the wrapper unless you want to code different behaviour for different tables.

At this time I have no data to determine which method is more likely to produce the better execution plan. However, when performance problems occur in production they are instinctively routed to the DBA, who is likely to have difficulty introducing a code change at short notice. Dynamic Sampling has some clear advantages.

*My thanks to Wolfgang Brietling (<http://www.centrexcc.com/>) for his comments in the original thread on Oracle-L, and his input into this article.*

*David Kurtz ([david.kurtz@go-faster.co.uk](mailto:david.kurtz@go-faster.co.uk)) is a performance specialist working with Oracle RDBMS and Enterprise PeopleSoft systems. He is a regular presenter at Oracle conferences, and is the author of 'PeopleSoft for the Oracle DBA', published by Apress ([www.psftdba.com](http://www.psftdba.com)).*