TECHNICAL NOTE

MANAGING OPTIMIZER STATISTICS FOR PEOPLESOFT ON ORACLE

Prepared By David Kurtz, Go-Faster Consultancy

Technical Note

http://www.go-faster.co.uk/GFCPSSTATS11.manual.pdf

Monday 12 July 2021

(E-mail: david.kurtz@go-faster.co.uk, telephone +44-7771-760660)

File: gfcpsstats11.docx, 12 July 2021

Contents

Introduction 3
- Maintenance Window 3
- Table Statistics Preferences 3
- Statistics on Temporary Working Storage Tables 4

Managing Statistics on PeopleSoft on Oracle with GFCPSSTATS package 5
- Locking and Deleting Statistics on Working Storage Tables 5
- Oracle's PSCBO_STATS package 6

Implementation 7

Meta-Data 8
- PS_GFC_STATS_OVRD 8
- Example Meta-Data 10
  - Example Time & Labor Meta-Data 10
- DDL Models 12
  - Simple 12
  - Full 12

Packaged Procedure GFCPSSTATS11 13
- PS_STATS Procedure 13
- REFRESH_STATS Procedure 14
- SET_TABLE_PREFS Procedure 14

SET_RECORD_PREFS Procedure 15
GENERATE_METADATA Procedure 15
Triggers 16
Required Privileges 16
GFC_STATS_OVRD_METADATA 16
GFC_STATS_OVRD_CREATE_TABLE 17
Delivered Files 17

Introduction

This document sets out a general approach for managing the collection of cost-based optimizer statistics in an Oracle database supporting a PeopleSoft system. The guiding principles are:

- Tell the optimizer the truth about your data. If you don't you can't expect it to make the right decisions. Although there cases that even where you do, it still won't make the right decisions.
- Manage the database as you would any other Oracle database in your estate. PeopleSoft.
- Work with the standard database mechanisms and features to achieve this.

Maintenance Window

The standard Oracle maintenance window job should be used to refresh stale statistics.

- By default the maintenance window runs from 10pm to 6am on weekdays, and throughout the weekends. This window is suitable for many systems; however, it is perfectly valid to adjust it to match times when a system is relatively quiet.
- Occasionally, it may be advantageous to add other short maintenance windows to keep after tables whose statistics have become stale and it is affecting performance.
- It may also be necessary to create custom jobs to maintain statistics for specific tables. For example, the maintenance windows may not be long enough to deal with any exceptionally large tables. Statistics should be locked on any table that is not to be managed by the maintenance window processes, and then statistics should be collected with the FORCE=>TRUE option to override the lock.

From 19c, on an engineered system, the High-Frequency Automatic Statistics Collection job can also be configured. This can be thought of as a continuous maintenance window, if enabled, then by default it runs every 15 minutes, for a maximum of 60 minutes, and collects statistics on the stalest tables first.

Table Statistics Preferences

In 11g, Oracle introduced the concept of table statistics preferences to declaratively specify how statistics should be collected, and they have expanded these in subsequent releases. There are global preferences that define default behaviours. Oracle delivers certain values, but they can be changed. Then preferences can also be specified on individual tables to describe how statistics are to be collected for that table. Preferences ensure that statistics are always collected consistently for a particular table, whoever collects them, and whenever they are collected.

In any application, and PeopleSoft is no exception, the aim should be to call the DBMS_STATS procedures with the minimum of parameters, to use the defaults wherever possible, and to use table statistics prefernces to override them where necessary.

Parameters can be specified in DBMS_STATS procedures to override the preferences. However, from Oracle 19 there is a new preference PREFERENCE_OVERRIDES_PARAMETER. If enabled DBMS_STATS parameters are ignored. If any parameters are specified in application code they are simply ignored at run time, and the table statistics preference is used instead. This allows the DBA to take back control over statistics collection from the developers. Where the application specifies valid parameters, such as METHOD_OPT, DEGREE or GRANULARITY this should be migrated to table statistics preferences. This preference is not enabled by default, but I strongly recommend setting it globally.

A common problem that I see with statistics collection is collecting them at with a non-default sample size is specified in application code. This will cause DBMS_STATS to cease using the new hash based number-of-distinct-values algortithm, introduced in 11g. Instead, it will revert to the legacy behaviour of sorting the data which takes a lot longer and uses more memory. From 12c, non-default sample size will also prevent generation hybrid histograms, and Oracle will revert to height-balanced histograms for columns with more than distinct values than the number of buckets.

Very simply, from 11g sample size should never be specified. From 19c, PREFERENCE_OVERRIDES_PARAMETER should be specified. It will solve this problem if you already have it, and if not, prevent it from occurring.

Statistics on Temporary Working Storage Tables

PeopleSoft processes often write data to temporary working storage tables and then use that data later in the same process. If the statistics on that table are not correct, it is quite likely to result poor performance when the optimizer makes an incorrect decision.

Some delivered processes collect statistics themselves after populating a working storage table. Over the years, PeopleSoft has added this behaviour to more processes. In Application Engine the %UpdateStats macro is used. It calls the command specified in the DDLMODEL for the current platform.

Application Engine processes generally truncate the data in non-shared instances of temporary records at the start of the process. Truncate does not affect statistics. The data is usually left behind in the table after the process has finished. If the process does not collect statistics, then the maintenance window process will collect statistics and the next time the process runs it will repopulate the table and the optimizer will have work out how to query new data with new bind variable and literal values and old statistics.

Therefore, we don't want the maintenance windows processes to gather statsitics on working storage tables. Statistics on PeopleSoft temporary records, and certain regular table records should be locked and deleted. Locked to omit them from the maintenance jobs, and deleted so that no statistics are present unless gathered by the process that populates them. If statistics are not present, Oralce will use Optimizer Dynamic Sampling to determine the execution plan.

Managing Statistics on PeopleSoft on Oracle with GFCPSSTATS package

Locking and Deleting Statistics on Working Storage Tables

When it comes to applying this principle in PeopleSoft, we have a few challenges to overcome, but nothing that is impossible.

When using PeopleSoft's Application Designer to migrate it is typical to rebuild a table in order to add or change a column. Any related objects will be lost. Application Designer will rebuild the indexes. However, auditing triggers will be lost. The same is true for table statistics preferences that are lost when a table is dropped. So we need to hold the preferences as meta-data in a table. Then we can apply the preferences to the tables as they are created using a DDL trigger.

PeopleSoft temporary records can correspond to many tables that are used as non-shared temporary working storage tables. The same table preferences will need to be applied to all the temporary table instances. So the meta-data needs to be held for each record.

The *%UpdateStats* macro is used to collect statistics during Application Engine programs. This uses the DDL model to call the *dbms_stats* PL/SQL procedure. The simplest approach would be to change the delivered DDL model to remove the *sample_size* and *method_opt* parameters but to specify *force=>TRUE*. This can be implemented with the script ddlora-gfcstats11-simple.sql. This may be the preferred option unless you need to supress statistics collection on particular tables (e.g.some of the temporary working storage records used in Time & Labour processing).

```
REM ddlora-gfcpsstats11-simple.sql
spool ddlora-gfcpsstats11-simple

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT = 'DBMS_STATS.GATHER_TABLE_STATS (ownname=>[DBNAME], tabname=>[TBNAME], force=>TRUE);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

If you do need this level of control then the DDL model should be changed to call the *gfcpsstats11* package.

```
REM ddlora-gfcpsstats11.sql
spool ddlora-gfcpsstats11

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT = 'gfcpsstats11.ps_stats(p_ownname=>[DBNAME],p_tabname=>[TBNAME]);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

I have retained the functionality developed in the original *wrapper* package to optionally suppress collection of statistics, or only to collect statistics when they are stale.

In the case of COBOL programs, the %UpdateStats macro is put into stored statements. These can be changed to call the same PL/SQL package as the DDL models. However, since PT8.55, PeopleSoft COBOL no longer uses this approach, but uses the DDL models.

I generally recommend increasing the setting of the Oracle initialisation parameter OPTIMIZER_DYNAMIC_SAMPLING from the default of 2 to 4. However, I have experienced some problems with nVision where there can be hundreds of predicates. In that specific case I would revert to the default level of 2.

I have created a new packaged procedure GFCPSSTATS11, it is based on the wrapper package that I produced for 10g and has evolved with subsequent versions of the database.

Oracle's PSCBO_STATS package

I can't discuss collecting statistics for PeopleSoft without discussing Oracle's CBO_STATS package. Oracle published document 1322888.1 "*pscbo_stats - Improving Statistics in Oracle RDBMS for PeopleSoft Enterprise*". It takes a similar approach to the package I proposed in the second edition of PeopleSoft for the Oracle DBA. A PL/SQL package is used to collect statistics. A number of tables control whether and how statistics are collected on each record. The package is also intended to be used to collect schema-wide statistics.

The PSCBO_STATS package is a valiant attempt to solve a genuine problem, and it has continued to evolve since its initial release. However, I have a number of objections to it.

- It is fundamentally a 10g solution. It does not use 11g table preferences.
- It does use the Oracle automatic sample size in 11g if histograms are not to be collected. Otherwise, it defaults to the previous behaviour of either using 100% sample size for when called by %UpdateStats with the 'high' sample size, or a variety of fixed sampe sizes based on internal rules and the number of rows in the table.
    - In 11g, automatic sample size produces better values, more quickly, and if necessary a specific sample size can be set with a table preference.
    - In 12c, automatic sample size must be used to get Top-N histograms.
- The package contains a procedure that collects statistics on all objects in the schema that also refreshes statistics that are not stale but which have not been refreshed for a period of time determined by the size of the table.
    - In 11g, this can be handled by setting a table specific stale threshold. A low threshold might be appropriate for larger tables.
- When it collects histograms it always sets the maximum bucket size of 254. This may not always be desirable for height balanced histograms.
    - In 11g, the collection of histograms can be controlled via the METHOD_OPT table preference.
- There is no support for collecting aggregated or incremental statistics on partitioned objects.
    - In 11g, table preferences for granularity and incremental statistics can be set.

I think that the 11g table preferences offer better and finer control over the collection of statistics.

APPENDIX

Implementation

All of the scripts are available on Github at https://github.com/davidkurtz/gfcpsstats

1. Using script gfcpsstats11.sql.
   a. Create metadata table PS_GFC_STATS_OVRD[1], and index,
      i. You may also choose to create a record in PeopleSoft Application Designer to correspond to this table.
   b. Create PL/SQL packaged procedure GFCPSSTATS11.
   c. Create trigger GFC_STATS_OVRD_METADATA
   d. Create trigger GFC_LOCKTEMPRECSTATS.
2. Update PeopleSoft DDL models 4 and 5 for Oracle to either one of the two following options
   a. Simply call *dbms_stats.gather_table_stats()* procedure with minimal parameters.
      i. Run SQL script ddlora-gfcstats11-simple.sql

```
REM ddlora-gfcpsstats11-simple.sql
REM (c) Go-Faster Consultancy 2008-21
spool ddlora-gfcpsstats11-simple

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT = 'DBMS_STATS.GATHER_TABLE_STATS (ownname=>[DBNAME],
tabname=>[TBNAME], force=>TRUE);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

   b. Or, if it is necessary to control whether stats are collected per record,call the new package instead of *dbms_stats.gather_table_stats* using
      i. Either data mover script ddlora-gfcpsstats11.dms
      ii. Or SQL script ddlora-gfcstats11.sql

```
REM ddlora-gfcpsstats11.sql
REM (c) Go-Faster Consultancy 2008-21
spool ddlora-gfcpsstats11

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT =
'gfcpsstats11.ps_stats(p_ownname=>[DBNAME],p_tabname=>[TBNAME]);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

Now, other scripts to lock statistics from https://github.com/davidkurtz/psscripts can be implemented

3. Run locktemprecstats.sql to lock and delete statistics on all PeopleSoft temporary records.
4. This step is optional: Run deltempstats.sql to create a trigger on table PS_AETEMPTBLMGR to lock and delete statistics on a temporary table as it is allocated to an Application Engine program.

---

[1] All scripts are delivered with explicit references to the default PSOWNERID schema, SYSADM. If you use a difference name for your PeopleSoft schema, you will have to globally search and replace this value.

Meta-Data

PS_GFC_STATS_OVRD

This table contains the meta-data that will be used to create the table preferences. It is keyed on record name. A record will only have table preferences created if a row exists in this table.

This table also controls whether

- the PS_STATS program in the GFCPSSTATS11 procedure will refresh statistics on a table.
- the statistics on the table(s) corresponding to the record are locked

The table is structured in line with PeopleTools so that it could also be created by defining a record in Application Designer and creating it like any other table in a PeopleSoft system.

Column	Datatype	Description
RECNAME*	VARCHAR2(15)	PeopleSoft Record Name
GATHERS_STATS	VARCHAR2(1)	This parameter controls the behaviour of the PS_STATS procedure (see page 13) that is designed to be called from the DDL models that are invoked by the *%UpdateStats* macro G=Gather statistics. This is effectively the same behaviour as the default DDL model. R=Refresh statistics. The *ps_stats* package will only gather statistics if they are marked as stale on the database. N=Don't Gather statistics
ESTIMATE_PERCENT	VARCHAR2(30)	Value passed to ESTIMATE_PERCENT table preference. If blank, any existing table override will be removed and Oracle will revert to default behaviour.
BLOCK_SAMPLE	VARCHAR2(1)	This does not correspond to a table preference. This value is passed to dbms_stats.gather_table_stats when it called by PS_STATS program. Y=TRUE N=FALSE
METHOD_OPT	VARCHAR2(1000)	Value passed to METHOD_OPT table preference. If blank, any existing table override will be removed and Oracle will revert to default

		behaviour.
GRANULARITY	VARCHAR2(30)	Value passed to GRANULARITY table preference. If blank, any existing table override will be removed and Oracle will revert to default behaviour.
INCREMENTAL	VARCHAR2(5)	Value passed to INCREMENTAL table preference. If blank, any existing table override will be removed and Oracle will revert to default behaviour.
STALE_PERCENT	NUMBER	Value passed to STALE_PERCENT table preference. If blank, any existing table override will be removed and Oracle will revert to default behaviour.
PREF_OVER_PARAM	VARCHAR2(1)	Value passed to PREFERENCE_OVERRIDES_PARAM preferences. If blank, any existing table override will be removed and Oracle will revert to default behaviour. New in Oracle 12.2. Will not be set in lower versions.
LOCK_STATS	VARCHAR2(1)	Controls whether statistics on table to be locked. By default type 0 records are unlocked, type 7 records are locked. Y=Locked N=Unlock

Example Meta-Data

Example Time & Labor Meta-Data

This the meta-data I defined for a Time & Labor system where excessive amounts of time were spent collecting statistics during TL_TIMEADMIN, the main T&L calculation process. We first tried to suppress statistics collection in all Application Engine processes and use only Optimizer Dymanic Sampling. We found that we still needs statistics, though not histograms, on certain tables.

RECNAME	GATHER_STATS	METHOD_OPT
TL_IPT1	G	FOR ALL COLUMNS SIZE 1
TL_MTHCD	G	FOR ALL COLUMNS SIZE 1
TL_PMTCH1_TMP	G	FOR ALL COLUMNS SIZE 1
TL_PMTCH2_TMP	G	FOR ALL COLUMNS SIZE 1
TL_PMTCH_TMP2	G	FOR ALL COLUMNS SIZE 1
TL_PROF_WRK	G	FOR ALL COLUMNS SIZE 1
TL_RESEQ2_WRK	G	FOR ALL COLUMNS SIZE 1
TL_RESEQ5_WRK	G	FOR ALL COLUMNS SIZE 1
TL_WRK01_RCD	G	FOR ALL COLUMNS SIZE 1
WRK_SCHRS_TAO	G	FOR ALL COLUMNS SIZE 1
TL_FRCS_PYBL_TM	R	
TL_ST_PCHTIME	R	
TL_VALID_TR	R	
All other temporary records in TL_TIMEADMIN	N	

This data can be loaded into the meta-data table with the script *gfcpsstats11_metadata.sql*.

```
REM gfcpsstats11_metadata.sql
REM (c) Go-Faster Consultancy 2008-2012
-----------------------------------------------------------------------------------------
--this script containts sample meta data table designed to minimize stats collection in TL
--processing these tables need statistics, but not histograms.  Optimizer Dynamic Sample
--is sufficient for the other temporary records used by TL_TIMEADMIN
--
--this list is just a suggestion - it worked at on one system - YOUR MILEAGE MAY VARY
--
--25.04.2014: On 11g you might be able to disable stats collection on TL_WRK01_RCD, but you still need stats on
TL_PROF_WRK
-----------------------------------------------------------------------------------------
clear screen
set echo on serveroutput on lines 100 wrap off
spool gfcpsstats11_metadata

REM DELETE FROM ps_gfc_stats_ovrd;

INSERT INTO ps_gfc_stats_ovrd (recname, gather_stats, estimate_percent, block_sample, method_opt, degree, granularity,
incremental, stale_percent, pref_over_param, lock_stats)
SELECT          recname,'G',' ',' ','FOR ALL COLUMNS SIZE 1',' ',' ',' ',0,' ',' '
FROM    psrecdefn a
WHERE   rectype IN(0,7)
AND     recname IN ('TL_IPT1'       ,'TL_MTCHD'     ,'TL_PMTCH1_TMP' ,'TL_PMTCH2_TMP' ,'TL_PMTCH_TMP1'
                   ,'TL_PMTCH_TMP2' ,'TL_PROF_LIST' ,'TL_PROF_WRK'   ,'TL_RESEQ2_WRK' ,'TL_RESEQ5_WRK'
                   ,'TL_WRK01_RCD'  ,'WRK_SCHRS_TAO')
AND NOT EXISTS(
      SELECT 'x'
      FROM        ps_gfc_stats_ovrd b
      WHERE       b.recname = a.recname);


INSERT INTO ps_gfc_stats_ovrd (recname, gather_stats, estimate_percent, block_sample, method_opt, degree, granularity,
incremental, stale_percent, pref_over_param, lock_stats)
SELECT          recname,'R',' ',' ',' ',' ',' ',' ',' ',0,' ',' '
FROM    psrecdefn a
WHERE   rectype IN(0,7)
AND     recname IN('TL_FRCS_PYBL_TM', 'TL_ST_PCHTIME', 'TL_VALID_TR')
AND NOT EXISTS(
      SELECT 'x'
      FROM        ps_gfc_stats_ovrd b
      WHERE       b.recname = a.recname)
;

INSERT INTO ps_gfc_stats_ovrd (recname, gather_stats, estimate_percent, block_sample, method_opt, degree, granularity,
incremental, stale_percent, pref_over_param, lock_stats)
SELECT          DISTINCT recname,'N',' ',' ',' ',' ',' ',' ',' ',0,' ',' '
FROM  PSAEAPPLTEMPTBL a
WHERE a.ae_applid = 'TL_TIMEADMIN'
AND NOT EXISTS(
      SELECT 'x'
      FROM        ps_gfc_stats_ovrd b
      WHERE       b.recname = a.recname);

commit;
```

DDL Models

Simple

If it is not necessary to be able to suppress statistics calls for certain records, then it may be simpler to call the Oracle *dbms_stats* package directly, but without any parameters, and with the force option.

```
REM ddlora-gfcpsstats11-simple.sql
spool ddlora-gfcpsstats11-simple

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT = 'DBMS_STATS.GATHER_TABLE_STATS (ownname=> [DBNAME], tabname=>[TBNAME], force=>TRUE);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

Full

This data mover script replaces DDL model 4 and 5 which are called from the Application Engine %UpdateStats macro. Note that both models are the same so it does not matter if the HIGH or LOW parameter is specified.

Calling the *gfcpsstats11* package in the DDL model allows metadata to determine whether the statistics call is supressed or modified for certain records.

```
--  ******************************************************************
--  ddlora-gfcpsstats11.dms (c) Go-Faster Consultancy 2012
--  ReLoads the PeopleTools DDL tables for Analyze statements for Oracle
--  calling gfcpsstats11 package
--  ******************************************************************

SET LOG DDLORA.LOG;

DELETE FROM PSDDLMODEL
WHERE PLATFORMID=2
AND STATEMENT_TYPE IN (4,5)
;

INSERT INTO PSDDLMODEL (
STATEMENT_TYPE,
PLATFORMID,
SIZING_SET,
PARMCOUNT,
MODEL_STATEMENT)
VALUES(
:1,
:2,
:3,
:4,
:5)
\
$DATATYPES NUMERIC,NUMERIC,NUMERIC,NUMERIC,CHARACTER
4,2,0,0,$long
gfcpsstats11.ps_stats(p_ownname=>[DBNAME],p_tabname=>[TBNAME],p_verbose=>TRUE);
//
5,2,0,0,$long
gfcpsstats11.ps_stats(p_ownname=>[DBNAME],p_tabname=>[TBNAME],p_verbose=>TRUE);
//
/
```

Alternatively, this change can be made by SQL statement

```
REM ddlora-gfcpsstats11.sql
spool ddlora-gfcpsstats11

UPDATE PSDDLMODEL
SET    MODEL_STATEMENT = 'gfcpsstats11.ps_stats(p_ownname=>[DBNAME],p_tabname=>[TBNAME]);'
WHERE  PLATFORMID=2
AND    STATEMENT_TYPE IN (4,5)
;

spool off
```

Packaged Procedure GFCPSSTATS11

The GFCPSSTATS11 package contains a number of public procedures that can be called extermally.

- PS_STATS
- REFRESH_STATS
- SET_TABLE_PREFS
- SET_RECORD_PREFS
- GENERATE_METADATA

NB: This procedure calls the PSFTAPI package; www.go-faster.co.uk/scripts.htm#psftapi.sql.

PS_STATS Procedure

This procedure is designed to be called from the DDL model for %UpdateStats by Application Engine processes and from within Cobol Stored Statements. It collects statistics on the named table according to the meta-data and table preferences.

Syntax

```
ps_stats
(p_ownname      IN VARCHAR2 /*owner of table*/
,p_tabname      IN VARCHAR2 /*table name*/
,p_verbose      IN BOOLEAN DEFAULT FALSE /*if true print SQL*/
);
```

Parameters

Parameter	Data Type	Description
p_ownname	VARCHAR2	Table Owner
p_tabname	VARCHAR2	Table Name
p_verbose	BOOLEAN	If true print debug messages

REFRESH_STATS Procedure

This procedure refreshs unlocked stale statistics on physical tables, partitions and subpartitions and any global statistics.

Syntax

```
set_defaults(
(p_ownname      IN VARCHAR2 /*owner of table*/
,p_tabname      IN VARCHAR2 /*table name*/
,p_block_sample IN BOOLEAN  DEFAULT FALSE /*if true block sample stats*/
,p_force        IN BOOLEAN  DEFAULT FALSE
,p_verbose      IN BOOLEAN  DEFAULT FALSE /*if true print SQL*/
);
```

Parameters

Parameter	Data Type	Description
p_ownname	VARCHAR2	Table Owner
p_tabname	VARCHAR2	Table Name
p_block_sample	BOOLEAN	Same as block_sample from *dbms_stats.gather_table_stats*. Whether or not to use random block sampling instead of random row sampling.
p_force	BOOLEAN	Same as force from *dbms_stats.gather_table_stats*. Gather statistics even if locked.
p_verbose	BOOLEAN	If true print debug messages

SET_TABLE_PREFS Procedure

This procedure sets table preferences on named tables according to the meta-data in PS_GFC_STATS_OVRD.

Syntax

```
set_table_prefs
(p_tabname      IN VARCHAR2 /*table name*/
,p_recname      IN VARCHAR2 DEFAULT NULL /*record of table if known*/
);
```

Parameters

Parameter	Data Type	Description
p_tabname	VARCHAR2	Table Name
p_recname	VARCHAR2	PeopleSoft Record Name

SET_RECORD_PREFS Procedure

This procedure to set table preferences on tables relating to named record to values specified in the meta-data. If there is no meta-data for the record the preferences are removed.

Syntax

```
set_record_prefs
(p_recname          IN VARCHAR2 /*record name*/
);
```

Parameters

Parameter	Data Type	Description
p_recname	VARCHAR2	PeopleSoft Record Name

GENERATE_METADATA Procedure

This procedure updates PS_GFC_STATS_OVRD so that the meta-data matches the actual table preferences.

Syntax

```
generate_metadata;
```

Triggers

Two triggers have been used to automate the application of table preferences.

Required Privileges

The triggers require that the following privileges are granted explicitly to the PeopleSoft owner schema, by default SYSADM.

```
REM gfcpsstats11_privs.sql
REM (c) Go-Faster Consultancy 2008-2012

GRANT EXECUTE ON dbms_scheduler TO sysadm;
GRANT CREATE JOB TO sysadm;
```

GFC_STATS_OVRD_METADATA

This trigger submits database jobs to reapply the table preferences when the metadata is updated and committed.

```
CREATE OR REPLACE TRIGGER gfc_stats_ovrd_metadata
AFTER INSERT OR UPDATE OF RECNAME, ESTIMATE_PERCENT, METHOD_OPT, DEGREE, GRANULARITY, INCREMENTAL, STALE_PERCENT
            OR DELETE ON ps_gfc_stats_ovrd
FOR EACH ROW
DECLARE
  l_cmd VARCHAR2(100) := '';
  l_jobno NUMBER;
BEGIN
  IF DELETING THEN
    dbms_job.submit(l_jobno,'gfcpsstats11.unset_record_prefs('''||:old.recname||''');');
  ELSE
    IF :new.recname != :old.recname THEN
      dbms_job.submit(l_jobno,'gfcpsstats11.unset_record_prefs('''||:old.recname||''');');
    END IF;
    dbms_job.submit(l_jobno,'gfcpsstats11.set_record_prefs('''||:new.recname||''');');
  END IF;
END gfc_stats_ovrd_metadata;
/
```

Setting and deleting table preferences include an implicit commit. Therefore, I have used DBMS_JOB to submit the jobs because the job is only executed when the trigger issues a commit. Thus, the procedure can just directly read the meta-data from the table. There will be a small lag between commiting meta-data changes, and the table preference being applied.

GFC_STATS_OVRD_CREATE_TABLE

This trigger applies table preferences to the table as it is created. A database job is used because otherwise the trigger fires while the table is being created but it exists.

```
CREATE OR REPLACE TRIGGER gfc_stats_ovrd_create_table
AFTER CREATE ON sysadm.schema
BEGIN
  IF ora_dict_obj_type = 'TABLE' THEN
    --submit one-time job to set table preferences as table will not have been created by time trigger runs
    sys.dbms_scheduler.create_job
    (job_name   => 'SET_PREFS_'||ora_dict_obj_name
    ,job_type   => 'PLSQL_BLOCK'
    ,job_action => 'BEGIN gfcpsstats11.set_table_prefs(p_tabname=>'''||ora_dict_obj_name||'''); END;'
    ,start_date => SYSTIMESTAMP --run job immediately
    ,enabled    => TRUE --job is enabled
    ,auto_drop  => TRUE --request will be dropped when complete
    ,comments   => 'Set table preferneces on table '||ora_dict_obj_owner||'.'||ora_dict_obj_name
    );
  END IF;
END;
/
```

Delivered Files

File Name	Description
gfcpsstats11.doc	This document.
gfcpsstats11.sql	Script that contains packaged procedure and triggers.
gfcpsstats11_privs.sql	Script to grant privileges to SYSADM schema.
gfcpsstats11_metadata.sql	Sample T&L metadata.
gfcpsstats11_testdata.sql	Script to test behaviour package.
ddlora-gfcpsstats11-simple.dms	Data Mover script to replace DDL Models 4 and 5 with commands to call *dbms_stats.gather_table_stats()* with no additional parameters other than FORCE=>TRUE
ddlora-gfcpsstats11.dms	Data Mover script to replace DDL Models 4 and 5 with commands to call *gfcpsstats11.ps_stats()*.