

# GFC\_ARCH\_MGMT: SEGMENT MANAGEMENT UTILITY PACKAGE

Prepared By David Kurtz, Go-Faster Consultancy Ltd.

Technical Note

Version 0.08

Tuesday 15 July 2014

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

File: gfc\_arch\_mgmt.20140715.docx, 15 July 2014

## Contents

Introduction.....	2
Overview of GFC_ARCH_MGMT .....	2
GFC_ARCH_MGMT package .....	3
LOCATE_TAB_PART.....	10
LOCATE_TAB_SUBPARTS.....	13
LOCATE_TAB_HASH_SUBPARTS .....	16
LOCATE_IND_PARTS .....	19
LOCATE_IND_SUBPARTS.....	21
LOCATE_LOB_PARTS .....	22
LOCATE_ALL_PARTS.....	24
Block Compression.....	27

## Introduction

This document discusses some of the additional segment management actions that can be conducted to reduce database size and I/O overhead during historical queries. These are separate to any archiving processes because they do not logically affect any data in the database. Rather, the partitioning affords the DBA team the opportunity to conduct these additional actions.

This document also explains the use of the GFC\_ARCH\_MGMT package that contains a number of programs to facilitate making these changes.

## Overview of GFC\_ARCH\_MGMT

The GFC\_ARCH\_MGMT package has been developed to manage aspects of partitioned objects, usually in conjunction with an archiving initiative.

PeopleSoft's Application Designer generates the DDL to create database tables and indexes. However, it does not support table partitioning<sup>1</sup>. Instead the GFC\_PSPART package is used to generate DDL to create and administer partitioned objects. The GFC\_ARCH\_MGMT package is driven by the same meta-data used by the GFC\_PSPART package that defines the partition structure, including physical attributes of the partitions.

Where physical partitioning of a table corresponds to periods of time, there are often opportunities to compress data that has become static, exchange partitions into an archive, and purge data by dropping partitions. Sometimes segments may be moved to a different tablespace. The GFC\_ARCH\_MGMT package has been developed to assist with these operations.

- the generally one-off task of moving segments to new tablespaces
- the regular on-going task of setting compression on segments as specified by the meta-data

Although table compression will reduce the size of table segments, it will leave large amounts of free space in the middle of a data file<sup>2</sup>. In order to reduce the overall database size, we need to be able move that free space to end of the data file and trim it off by resizing the data file. Another package GFC\_DEFRAG (see ), that is separately documented, has been created to assist with that task.

---

<sup>1</sup> This is because the implementation is platform specific. However, the PeopleSoft Red Paper on performance actively recommends the use of partitioning without explaining how to do it. Therefore, the use of partitioning does not invalidate product support.

From PeopleTools 8.51, Application Designer will attempt to preserve existing partitioning on Oracle by using the DBMS\_METADATA package.

<sup>2</sup> A favourite analogy for this is that of a swiss cheese with holes. The holes take up space in the cheese. We could reduce the overall size of the cheese, by filling the holes up with cheese taken from the outside. This is neither possible nor desirable with a real physical piece of Emmental, but it is with a data file!

## GFC\_ARCH\_MGMT package

This package contains various programs that dynamically generate and execute DDL to move segments to appropriate tablespaces and set store options as defined in the meta-data for the GFC\_PSPART package.

NB: This package was originally designed to be owned by SYS. This is because it requires a number of privileges that it would be unsafe to grant to SYSADM. It also resulted in much tighter control over the use of the package which is a utility for the DBAs.

The package contains the following sub-programs:

- LOCATE\_TAB\_PARTS (see page 3)
- LOCATE\_TAB\_SUBPARTS (see page 13)
- LOCATE\_TAB\_HASH\_SUBPARTS (see page 16)
- LOCATE\_IND\_PARTS (see page 19)
- LOCATE\_IND\_SUBPARTS (see page 21)
- LOCATE\_ALL\_PARTS (see page 22)

Ultimately all the LOCATE procedures will need to be run for at least some tablespaces both as part of the initial archiving and on an ongoing basis as each partition is compressed and archived. Perhaps the easiest way will be to schedule LOCATE\_ALL\_PARTS to run without any parameters for all the metadata and see what it picks up.

## Privileges

If the GFC\_ARCH\_MGMT package is not owned by SYS, it will require the following privileges be granted explicitly to its owner.

```

-----
--
-- script      : gfc_arch_mgmt_privs.sql
--
-- created by   : DMK
-- creation date : 19.10.2012
--
-- description  : script to grant privileges required by SYSADM owned gfc_ARCH_MGMT package
--
-- development & maintenance history
--
-- date        author          version  description
-----
-- 19.10.2012  DMK                1.01    initial version
-----

set echo on
spool gfc_arch_mgmt_privs
GRANT SELECT ON sys.dba_tablespace TO sysadm;
GRANT SELECT ON sys.dba_segments TO sysadm;
GRANT SELECT ON sys.dba_tables TO sysadm;
GRANT SELECT ON sys.dba_indexes TO sysadm;
GRANT SELECT ON sys.dba_tab_partitions TO sysadm;
GRANT SELECT ON sys.dba_ind_partitions TO sysadm;
GRANT SELECT ON sys.dba_lob_partitions TO sysadm;
GRANT SELECT ON sys.dba_tab_subpartitions TO sysadm;
GRANT SELECT ON sys.dba_ind_subpartitions TO sysadm;
GRANT EXECUTE ON sys.dbms_scheduler TO sysadm;
spool off

```

[gfc\\_arch\\_mgmt\\_privs.sql](#)

## Implementation

The following files are prerequisites:

1. psftapi.sql: <http://www.go-faster.co.uk/scripts.htm#psftapi.sql>
2. Psft\_ddl\_lock: [http://www.go-faster.co.uk/scripts.htm#psft\\_ddl\\_lock.sql](http://www.go-faster.co.uk/scripts.htm#psft_ddl_lock.sql)

The following files are provided.

1. *gfc\_arch\_mgmt\_privs.sql*: grants privileges required by package



*gfc\_arch\_mgmt\_privs.sql*

2. *gfc\_arch\_mgmt\_pkg.sql*: creates package header and body.



*gfc\_arch\_mgmt\_pkg.sql*

3. *gfc\_arch\_mgmt\_pkg\_sample.sql*: script containing a number of sample commands.



*gfc\_arch\_mgmt\_pkg\_sample.sql*

4. *gfc\_arch\_mgmt\_locate\_all\_parts.sql*: Simplest form of command that will run package for all objects



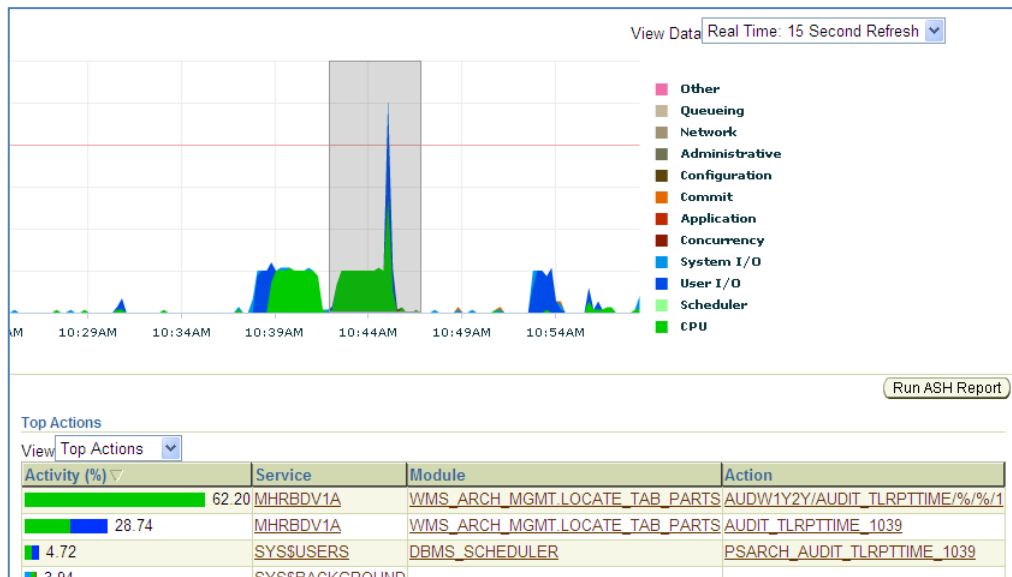
gfc\_arch\_mgmt\_locate\_all\_parts.sql

**Instrumentation**

Each program in the package uses DBMS\_APPLICATION\_INFO to set the module and action. These values can be used to track the progress of the package in Oracle Enterprise Manager etc.

- Module is set to the name of the package followed by the name of the program. For example, while the LOCATE\_TAB\_PARTS package is running it would be set to GFC\_ARCH\_MGMT.LOCATE\_TAB\_PARTS.
- Action is initially set to the list of parameters passed to the program. As the programs move on to rebuilding segments the action is updated to the name of the segment.

So in the following example screenshot from OEM<sup>3</sup>, I can see that LOCATE\_TAB\_PARTS was run at 10:44 for a single partition in record AUDIT\_TLRPTTIME. It selected partition AUDIT\_TLRPTTIME\_1039. You can also see the database job to run ti collect statistics on the segment AUDIT\_TLRPTTIME\_1039.



<sup>3</sup> Taken from a customised version of this package on a customer system.

## Logging

The package logs the start and end of public packages and all dynamically generated DDL commands.

- i) All logging is emitted via the Oracle *dbms\_output* package to the client terminal (if serveroutput is enabled).

```
14:22:44 23.10.2012:GFC_ARCH_MGMT.LOCATE_TAB_PARTS(part_id=>%,recname=>%,partname=>%,tabl
espace=>%,updateind=>TRUE,statsjob=>TRUE,max_parts=>1,testmode=>FALSE)
15:06:51 23.10.2012:Partition AUDIT_XXXXXXXX_1039:ENABLED(775%,497117 rows,2147
blocks) -> COMPRESS
15:06:51 23.10.2012:ALTER TABLE PSARCH.PS_AUDIT_XXXXXXXX MOVE PARTITION
AUDIT_XXXXXXXX_1039 TABLESPACE HRLARGE UPDATE INDEXES COMPRESS NOPARALLEL
15:07:13
23.10.2012:GFC_ARCH_MGMT.TAB_STATS_JOB(ownname=>PSARCH,tabname=>PS_AUDIT_XXXXXX
XX,partname=>AUDIT_XXXXXXXX_1039,granularity=>PARTITION,testmode=>FALSE)
15:07:13 23.10.2012:BEGIN
sys.dbms_stats.gather_table_stats(ownname=>'PSARCH',tabname=>'PS_AUDIT_XXXXXXXX
',partname=>'AUDIT_XXXXXXXX_1039',estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE
,granularity=>'PARTITION',degree=>DBMS_STATS.DEFAULT_DEGREE,method_opt=>'FOR ALL
COLUMNS SIZE REPEAT',cascade=>TRUE); END;
15:07:13 23.10.2012:Submitted Job PSARCH_AUDIT_XXXXXXXX_1039: Collect
statistics on table partition PSARCH.PS_AUDIT_XXXXXXXX.AUDIT_XXXXXXXX_1039
15:07:13 23.10.2012:ALTER TABLE PSARCH.PS_AUDIT_XXXXXXXX NOPARALLEL
15:07:13 23.10.2012:GFC_ARCH_MGMT.LOCATE_TAB_PARTS completed

PL/SQL procedure successfully completed.
```

- ii) If the package is owned by SYS then logging can also emitted to the session trace file in the *user\_dump\_dest* directory using *dbms\_output.kswrtd*.

```

/app/oracle/admin/ABCDEFGH/udump/ABCDEFGH_ora_26240.trc
Oracle Database 10g Enterprise Edition Release 10.2.0.5.0 - 64bit Production
With the Partitioning, Data Mining and Real Application Testing options
ORACLE_HOME = /app/oracle/product/10.2.0.5/db_2
System name:      XXXXX
Node name:       AUDIT_XXXXXXXXXX
Release:        X.XX.XX
Version:        X
Machine:        XXXX
Instance name:  XXXXXXXX
Redo thread mounted by this instance: 1
Oracle process number: 54
Unix process pid: 26240, image: oracle@xxxxxxxx (TNS V1-V3)

*** 2012-10-23 14:22:44.453
*** ACTION NAME: (%%/%%/1) 2012-10-23 14:22:44.452
*** MODULE NAME: (GFC_ARCH_MGMT.LOCATE_TAB_PARTS) 2012-10-23 14:22:44.452
*** SERVICE NAME: (SYS$USERS) 2012-10-23 14:22:44.452
*** SESSION ID: (449.39849) 2012-10-23 14:22:44.452
*** 2012-10-23 14:22:44.453
GFC_ARCH_MGMT.LOCATE_TAB_PARTS(part_id=>%,recname=>%,partname=>%,tablespace=>%,updateind=>TRUE,statsjob=>TR
UE,max_parts=>1,testmode=>FALSE)
*** 2012-10-23 15:06:51.777
*** ACTION NAME: (AUDIT_XXXXXXXXX_1039) 2012-10-23 15:06:51.777
*** MODULE NAME: (GFC_ARCH_MGMT.LOCATE_TAB_PARTS) 2012-10-23 15:06:51.777
Partition AUDIT_XXXXXXXXX_1039:ENABLED(775%,497117 rows,2147 blocks) -> COMPRESS
ALTER TABLE PSARCH.PS_AUDIT_XXXXXXXXX MOVE PARTITION AUDIT_XXXXXXXXX_1039 TABLESPACE HRLARGE UPDATE INDEXES
COMPRESS NOPARALLEL
*** 2012-10-23 15:07:13.033
*** ACTION NAME: (PSARCH/PS_AUDIT_XXXXXXXXX/AUDIT_) 2012-10-23 15:07:13.033
*** MODULE NAME: (GFC_ARCH_MGMT.TAB_STATS_JOB) 2012-10-23 15:07:13.033
GFC_ARCH_MGMT.TAB_STATS_JOB(ownname=>PSARCH,tabname=>PS_AUDIT_XXXXXXXXX,partname=>AUDIT_XXXXXXXXX_1039,gran
ularity=>PARTITION,testmode=>FALSE)
BEGIN
sys.dbms_stats.gather_table_stats(ownname=>'PSARCH',tabname=>'PS_AUDIT_XXXXXXXXX',partname=>'AUDIT_XXXXXXXX
X_1039',estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE,granularity=>'PARTITION',degree=>DBMS_STATS.DEFAULT_D
EGREE,method_opt=>'FOR ALL COLUMNS SIZE REPEAT',cascade=>TRUE); END;
Submitted Job PSARCH_AUDIT_XXXXXXXXX_1039: collect statistics on table partition
PSARCH.PS_AUDIT_XXXXXXXXX.AUDIT_XXXXXXXXX_1039
*** ACTION NAME: (AUDIT_XXXXXXXXX_1039) 2012-10-23 15:07:13.043
*** MODULE NAME: (GFC_ARCH_MGMT.LOCATE_TAB_PARTS) 2012-10-23 15:07:13.043
ALTER TABLE PSARCH.PS_AUDIT_XXXXXXXXX NOPARALLEL
*** 2012-10-23 15:07:13.045
GFC_ARCH_MGMT.LOCATE_TAB_PARTS completed

```

- iii) The start and end of public procedures is emitted to the instance alert log also using using *dbms\_output.kswrdt*.

```

Tue Oct 23 14:11:13 BST 2012
14:11:13
23.10.2012:LOCATE_TAB_PARTS(part_id=>AUD%,recname=>%,partname=>%,tablespace=>%,updateind=>TRUE,statsjob=>TR
UE,max_parts=>1,testmode=>TRUE)
Tue Oct 23 14:12:48 BST 2012
14:12:48
23.10.2012:LOCATE_TAB_PARTS(part_id=>%,recname=>AUDIT_TLRPTTIME,partname=>%,tablespace=>%,updateind=>TRUE,s
tatsjob=>TRUE,max_parts=>1,testmode=>FALSE)
Tue Oct 23 14:14:18 BST 2012
14:14:18 23.10.2012:GFC_ARCH_MGMT.LOCATE_TAB_PARTS completed
Tue Oct 23 14:22:44 BST 2012
GFC_ARCH_MGMT.LOCATE_TAB_PARTS(part_id=>%,recname=>%,partname=>%,tablespace=>%,updateind=>TRUE,statsjob=>TR
UE,max_parts=>1,testmode=>FALSE)
Tue Oct 23 15:07:13 BST 2012
GFC_ARCH_MGMT.LOCATE_TAB_PARTS completed

```

### Parallelism

By default the package does not use parallelism when rebuilding objects because PSFT\_DDL\_LOCK fires in a separate session for recursive rebuilds of indexes. However, parallelism may be invoked by specifying the *p\_parallel* parameter.

By default, when a segment is built in parallel, if the degree of parallelism is not specified the number of parallel session is determined by the product of two Oracle parameters; *cpu\_count* and *parallel\_threads\_per\_cpu*.

However, excessive parallelism on objects with fewer extents than the degree of parallelism can result in space wastage because each parallel process writes separate extents. An objects will have at least as many extents as the degree of parallelism used during rebuilding. This effect is more severe in locally managed tablespaces with large uniform extent sizes.

The package calculates default parallelism and compares it to an estimate of the number of extents it expects in the rebuilt segment<sup>4</sup>. If the expected number of extents is less than the default degree of parallelism then the degree of parallelism is specified as equal to the expected number of extents, otherwise no degree of parallelism is specified.

```

*** MODULE NAME: (GFC_ARCH_MGMT.READ_INIT_PARAMS) 2013-01-18 18:24:42.314
cpu_count=4,parallel_threads_per_cpu=2,max_parallelism=8
...
Subpartition TL_RPTD_TIME_1149_01 : COMPRESS (102%,84916 rows,2644 blocks, 10%free, analyzed 16.03.2012
13:46:03) -> PCTFREE 0 PCTUSED 99 COMPRESS
ALTER TABLE SYSADM.PS_TL_RPTD_TIME MODIFY PARTITION TL_RPTD_TIME_1149 PCTFREE 0 PCTUSED 99 COMPRESS
est_blocks=2710,typical_compression=4.2,initial_extent=1048576,block_size=8192,max_parallelism=8,parallelis
m=5
ALTER TABLE SYSADM.PS_TL_RPTD_TIME MOVE SUBPARTITION TL_RPTD_TIME_1149_01 TABLESPACE TL2011M12TAB UPDATE
INDEXES PARALLEL 5

```

However, if the expected number of extents is only 1 then parallelism is suppressed.

<sup>4</sup> If the segment is being compressed the calculation assumes an arbitrary constant compression factor based on observation of actual compression achieved – this factor is currently hard-coded as 4.2.



```
ALTER TABLE SYSADM.PS_TL_RPTD_TIME MOVE SUBPARTITION TL_RPTD_TIME_1031_03 TABLESPACE TL2010M08TAB UPDATE  
INDEXES NOPARALLEL
```

### Optimizer Statistics

Table compression, of course, does not change the data in a table, but does pack it into fewer blocks. Rebuilding an object to introduce compression will also have the effect of removing any space left over from update or delete operations. The number of blocks in an object is fundamental to the Cost-Based Optimizer. Its purpose is primarily to calculate how many blocks it will have to retrieve for each possible execution plan, thus determining the cheapest. Therefore it is necessary to update statistics on a segment after rebuilding it so that the Optimizer can determine accurate costings.

The various programs in the package submit a database job (using the DBMS\_SCHEDULER package) to collect statistics. The use of the database scheduler permits this to be done concurrently and in background. While the database job collects statistics on the last segment, the package can move on to the next segment. This behaviour can be optionally suppressed with a parameter.

## LOCATE\_TAB\_PART

This program moves table range partitions that are either in a different tablespace to that specified in the meta-data, or which have different compression specified to that specified in the meta-data.

### Notes:

- The generated move DDL includes the PARALLEL keyword if the table is estimated to occupy more than 512 uncompressed blocks.
- The meta-data specifying tablespace and storage options for range partitions is in GFC\_PART\_RANGES.

### Syntax

```
PROCEDURE locate_tab_parts
  (p_part_id      VARCHAR2 DEFAULT '%'
  ,p_recname     VARCHAR2 DEFAULT '%'
  ,p_partname    VARCHAR2 DEFAULT '%'
  ,p_tablespace  VARCHAR2 DEFAULT '%'
  ,p_updateind   BOOLEAN   DEFAULT TRUE
  ,p_statsjob    BOOLEAN   DEFAULT TRUE
  ,p_parallel    BOOLEAN   DEFAULT FALSE
  ,p_max_parts   INTEGER   DEFAULT NULL
  ,p_testmode    BOOLEAN   DEFAULT FALSE);
```

### Parameters

Parameter	Description
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of partition in database. DBA_TAB_PARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_RANGES.TAB_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_updateind	If this parameter is true the ALTER TABLE ... MOVE PARTITION command includes the UPDATE INDEXES command. Otherwise any local index partitions or global indexes on the table partition will be invalidated. The default value of the parameter is FALSE.  Note: If parallelism is enabled, the parallelism will also cascade on into any index rebuild. However, parallel index rebuild will fail if the PSFT_DDL_LOCK trigger is still enabled.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the table partition and its indexes. The default value is TRUE.
p_parallel	If this parameter is true then the table rebuild will be done in parallel by adding the PARALLEL keyword, if it is false parallelism will be suppressed by adding the NOPARALLEL keyword to the table rebuild. The default value of the parameter is TRUE. If it is explicitly set to NULL neither keyword will be added.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> <sup>5</sup> .

<sup>5</sup> Remember to *SET SERVEROUTPUT ON* in SQL\*Plus

## Examples

The locate\_tab\_parts program can be run in a number of ways depending upon what you want to process.

- In test mode just the SQL commands for a single table

```
execute sysadm.gfc_arch_mgmt.locate_tab_parts(p_max_parts=>1,p_testmode=>TRUE);
```

- Process all partitions in tablespace that begin with GP

```
execute sysadm.gfc_arch_mgmt.locate_tab_parts(p_tablespace=>'GP%');
```

- Process one partition from table PS\_AUDIT\_TLRPTTIME

```
execute sysadm.gfc_arch_mgmt.locate_tab_parts(p_recname=>'AUDIT_TLRPTTIME',p_max_parts=>1);
```

- Process partition AUDIT\_TLRPTTIME\_1025

```
execute sysadm.gfc_arch_mgmt.locate_tab_parts(p_partname=>'AUDIT_TLRPTTIME_1025');
```

## LOCATE\_TAB\_SUBPARTS

This program moves subpartitions in tables that are range-list subpartitioned that are either in a different tablespace to that specified in the meta-data, or which have the wrong compression.

Notes:

- The generated move DDL includes the PARALLEL keyword if the table is estimated to occupy more than 512 uncompressed blocks.
- The subpartition must have optimiser statistics collected. If the number of rows reported on the subpartition is either null or zero, it will not be processed.

In 10g, the compression reported on DBA\_TAB\_SUBPARTITIONS is merely the value set at partition level, not the actual state of the subpartition. It is not until 11g that it is possible to directly set compression at sub-partition level. Therefore, the package calculates whether the sub-partition is actually compressed or not (see Appendix: Block Compression on page 27).

- If a subpartition compression is to be changed, then the compression attribute is first set on the partition, then the subpartition is moved, and finally the original compression attribute is restored to the partition.
- The meta-data specifying tablespace and storage options for list sub-partitions is in GFC\_PART\_LISTS.

### Syntax

```

PROCEDURE locate_tab_subparts
  (p_part_id      VARCHAR2 DEFAULT '%'
  ,p_recname     VARCHAR2 DEFAULT '%'
  ,p_partname    VARCHAR2 DEFAULT '%'
  ,p_subpartname VARCHAR2 DEFAULT '%'
  ,p_tablespace  VARCHAR2 DEFAULT '%'
  ,p_updateind   BOOLEAN  DEFAULT TRUE
  ,p_statsjob    BOOLEAN  DEFAULT TRUE
  ,p_parallel    BOOLEAN  DEFAULT FALSE
  ,p_max_parts   INTEGER  DEFAULT NULL
  ,p_testmode    BOOLEAN  DEFAULT FALSE);

```

**Parameters**

<b>Parameter</b>	<b>Description</b>
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of partition in database. DBA_TAB_SUBPARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_subpartname	Name of partition in database. DBA_TAB_SUBPARTITIONS.SUBPARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all subpartitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_LISTS.TAB_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_updateind	If this parameter is true the ALTER TABLE ... MOVE PARTITION command includes the UPDATE INDEXES command. Otherwise any local index partitions or global indexes on the table partition will be invalidated. The default value of the parameter is TRUE.  Note: If parallelism is enabled, the parallelism will also cascade on into any index rebuild. However, parallel index rebuild will fail if the PSFT_DDL_LOCK trigger is still enabled.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the table partition and its indexes. The default value is TRUE.
p_parallel	If this parameter is true then the table rebuild will be done in parallel by adding the PARALLEL keyword, if it is false parallelism will be suppressed by adding the NOPARALLEL keyword to the table rebuild. The default value of the parameter is FALSE. If it is explicitly set to NULL neither keyword will be added.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> .



## LOCATE\_TAB\_HASH\_SUBPARTS

This procedure is similar to *locate\_tab\_subparts*, but it applies only to composite partitioned tables that are range partitioned and hash-subpartitoned. It moves subpartitions that are either in a different tablespace to that specified in the meta-data, or which have the wrong compression.

Notes:

- The generated move DDL includes the PARALLEL keyword if the table is estimated to occupy more than 512 uncompressed blocks.
- The subpartition must have optimiser statistics collected. If the number of rows reported on the subpartition is either null or zero, it will not be processed.

In 10g, the compression reported on DBA\_TAB\_SUBPARTITIONS is merely the value set at partition level, not the actual state of the subpartition. It is not until 11g that it is possible to directly set compression at sub-partition level. Therefore, the package calculates whether the sub-partition is actually compressed or not (see Appendix: Block Compression on page 27).

- If a subpartition compression is to be changed, then the compression attribute is first set on the partition, then the subpartition is moved, and finally the original compression attribute is restored to the partition.
- There is no parameter to specify an individual subpartition because if the table is hash-subpartitioned it makes no sense to treat different hashes differently<sup>6</sup>.

There is no meta-data to specify tablespace for hash sub-partiton, and storage options cannot be specified explicitly for sub-partitions in Oracle 10g. The meta-data in GFC\_PART\_RANGES that specifyies tablespace and storage options is applied to hash subpartitions.

---

<sup>6</sup> Hash-partitioning can be used to produce a similar effect to list-partitioning by choosing particular data values in the partitioning column such that they produced particular hash values. Indeed, it was done on a number of PeopleSoft GP systems. However, since the introduction of list partitioning in Oracle 9.2 that technique is now obsolete.



**Syntax**

```
PROCEDURE locate_tab_hash_subparts
(
  p_part_id      VARCHAR2 DEFAULT '%'
  ,p_recname     VARCHAR2 DEFAULT '%'
  ,p_partname    VARCHAR2 DEFAULT '%'
  ,p_tablespace  VARCHAR2 DEFAULT '%'
  ,p_updateind   BOOLEAN  DEFAULT TRUE
  ,p_statsjob    BOOLEAN  DEFAULT TRUE
  ,p_parallel    BOOLEAN  DEFAULT FALSE
  ,p_max_parts   INTEGER   DEFAULT NULL
  ,p_testmode    BOOLEAN  DEFAULT FALSE) ;
```

**Parameters**

<b>Parameter</b>	<b>Description</b>
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of partition in database. DBA_TAB_SUBPARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_RANGES.TAB_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_updateind	If this parameter is true the ALTER TABLE ... MOVE PARTITION command includes the UPDATE INDEXES command. Otherwise any local index partitions or global indexes on the table partition will be invalidated. The default value of the parameter is TRUE.  Note: If parallelism is enabled, the parallelism will also cascade on into any index rebuild. However, parallel index rebuild will fail if the PSFT_DDL_LOCK trigger is still enabled.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the table partition and its indexes. The default value is TRUE.
p_parallel	If this parameter is true then the table move will be done in parallel by adding the PARALLEL keyword, if it is false parallelism will be suppressed by adding the NOPARALLEL keyword to the table rebuild. The default value of the parameter is FALSE. If it is explicitly set to NULL neither keyword will be added.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> .

## LOCATE\_IND\_PARTS

This program rebuilds index range partitions (on locally range-partitioned indexes) that are in a different tablespace to that specified in the meta-data.

Notes:

- Just as in the DDL scripts generated by PeopleTools Application Designer, the rebuild is always done in parallel, and then index is altered NOPARALLEL.
- The meta-data specifying tablespace and storage options for local index range partitions is in GFC\_PART\_RANGES

### Syntax

```
PROCEDURE locate_tab_parts
    (p_part_id      VARCHAR2 DEFAULT '%'
    ,p_recname      VARCHAR2 DEFAULT '%'
    ,p_partname     VARCHAR2 DEFAULT '%'
    ,p_tablespace  VARCHAR2 DEFAULT '%'
    ,p_statsjob     BOOLEAN  DEFAULT TRUE
    ,p_max_parts   INTEGER  DEFAULT NULL
    ,p_testmode    BOOLEAN  DEFAULT FALSE) ;
```

### Parameters

Parameter	Description
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of partition in database. DBA_IND_PARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_RANGES.IDX_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the index partition. The default value is TRUE.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> .

## LOCATE\_IND\_SUBPARTS

This program rebuilds index subpartitions (on locally range-partitioned and list-subpartitioned indexes) that are in a different tablespace to that specified in the meta-data.

Notes:

- Just as in the DDL scripts generated by PeopleTools Application Designer, the rebuild is always done in parallel, and then index is altered NOPARALLEL.
- The meta-data specifying tablespace and storage options for local index range partitions is in GFC\_PART\_LISTS

### Syntax

```
PROCEDURE locate_tab_parts
    (p_part_id      VARCHAR2 DEFAULT '%'
    ,p_recname      VARCHAR2 DEFAULT '%'
    ,p_partname     VARCHAR2 DEFAULT '%'
    ,p_tablespace  VARCHAR2 DEFAULT '%'
    ,p_statsjob     BOOLEAN  DEFAULT TRUE
    ,p_max_parts   INTEGER  DEFAULT NULL
    ,p_testmode    BOOLEAN  DEFAULT FALSE);
```

### Parameters

Parameter	Description
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of partition in database. DBA_IND_PARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_LISTS.IDX_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the index partition. The default value is TRUE.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> .

## LOCATE\_LOB\_PARTS

This program rebuilds partitioned tables where the lob segment or lob index segment are not in the tablespace specified for the table in the meta-data.

Notes:

- The name of the table is

### Syntax

```
PROCEDURE locate_tab_parts
    (p_part_id      VARCHAR2 DEFAULT '%'
    ,p_recname     VARCHAR2 DEFAULT '%'
    ,p_partname    VARCHAR2 DEFAULT '%'
    ,p_tablespace  VARCHAR2 DEFAULT '%'
    ,p_updateind   BOOLEAN   DEFAULT TRUE
    ,p_statsjob    BOOLEAN   DEFAULT TRUE
    ,p_max_parts   INTEGER   DEFAULT NULL
    ,p_testmode    BOOLEAN   DEFAULT FALSE) ;
```

### Parameters

Parameter	Description
p_part_id	ID of partitioning strategy. GFC_PART_RANGES.PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of the lob segment partition in database. DBA_TAB_PARTITIONS.PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
p_tablespace	Name of tablespace where partition should reside. GFC_PART_RANGES.TAB_TABLESPACE is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_updateind	If this parameter is true the ALTER TABLE ... MOVE PARTITION command includes the UPDATE INDEXES command. Otherwise any local index partitions or global indexes on the table partition will be invalidated. The default value of the parameter is TRUE
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the table partition and its indexes. The default value is TRUE.
p_max_parts	If specified, the number of partitions processed by the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> <sup>7</sup> .

<sup>7</sup> Remember to *SET SERVEROUTPUT ON* in SQL\*Plus

## LOCATE\_ALL\_PARTS

This program simply calls each of the other LOCATE% programs described in this document passing parameters through, in the following order:

1. LOCATE\_TAB\_PARTS (see page 5)
2. LOCATE\_TAB\_SUBPARTS (see page 13)
3. LOCATE\_TAB\_HASH\_SUBPARTS (see page 16)
4. LOCATE\_LOB\_PARTS (see page 22)
5. LOCATE\_IND\_PARTS (see page 19)
6. LOCATE\_IND\_SUBPARTS (see page 21)

This program rebuilds partitioned tables where the lob segment or lob index segment are not in the tablespace specified for the table in the meta-data.

### Syntax

```
PROCEDURE locate_tab_parts
    (p_part_id      VARCHAR2 DEFAULT '%'
    ,p_recname     VARCHAR2 DEFAULT '%'
    ,p_partname     VARCHAR2 DEFAULT '%'
    ,p_subpartname VARCHAR2 DEFAULT '%'
    ,p_tablespace  VARCHAR2 DEFAULT '%'
    ,p_updateind   BOOLEAN  DEFAULT TRUE
    ,p_statsjob    BOOLEAN  DEFAULT TRUE
    ,p_max_parts   INTEGER  DEFAULT NULL
    ,p_testmode    BOOLEAN  DEFAULT FALSE);
```



### Parameters

Parameter	Description
p_part_id	ID of partitioning strategy. PART_ID is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioning strategies.
p_recname	Name of PeopleSoft record to be generated. RECNAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitioned records on GFC_PART_TABLES.
p_partname	Name of the lob segment partition in database. PARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all partitions.
P_subpartname	Name of partition in database. SUBPARTITION_NAME is matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all subpartitions.
p_tablespace	Name of tablespace where partition should reside. Either TAB_TABLESPACE or IDX_TABLESPACE are matched through a LIKE operator to the value of this parameter. The default value of the parameter is '%' which will match all tablespaces.
p_updateind	If this parameter is true the ALTER TABLE ... MOVE command includes the UPDATE INDEXES command. Otherwise any local index partitions or global indexes on the table partition will be invalidated. The default value of the parameter is TRUE.
p_statsjob	If this parameter is true a job will be submitted to the database job scheduler to collect statistics on the table or index partition or subpartition. The default value is TRUE.
p_max_parts	If specified, the number of partitions processed by each of the program will be limited to the value of this parameter. The default value is null, in which case there is no limit.
p_testmode	If this parameter is TRUE then the package will not submit the DDL that it generates to the database, it will only be printed through <i>dbms_output</i> <sup>8</sup> .

<sup>8</sup> Remember to *SET SERVEROUTPUT ON* in SQL\*Plus



## A P P E N D I X

## Block Compression

Several programs in GFC\_ARCH\_MGMT attempt to calculate whether the content of a segment is actually compressed, as opposed to whether the compression attribute is set in the data dictionary. This

The following formula is used to calculate the ratio of the number of blocks that the table would occupy if not compressed to the number of blocks that are actually used.

$$\text{block fill ratio} = \frac{\text{num\_rows.avg\_row\_len}}{\text{blocks} \cdot \left(1 - \frac{\text{pct\_free}}{100}\right) \cdot \left(\text{block size} - \left(57 + 23 \cdot \text{ini\_trans} + \frac{2 \cdot \text{num\_rows}}{\text{blocks}} + 4\right)\right)}$$

If the block fill ratio calculated above is over 100% then the segment is probably compressed. If it is less than 100% it is probably not compressed.

It is possible that the statistics on a table might not have been recalculated since it was compressed, so the number of blocks in a segment is taken to be the lesser of the number of blocks reported by the statistics, and the actual number of blocks in the segment.

The formula estimates the overhead of the block structures as:

- 57 bytes for the block header
- 23 bytes per entry for the transaction header.
- 4 bytes for the table directory, because there is only ever one entry in a non-clustered block
- 2 bytes for each entry in the row directory.

There are a number of limitations to the above formula.

- The formula does not apply to blocks in a clustered table.
- The estimate of the block header overhead makes no allowance for:
  - creation of addition transaction header entries as blocks are used,
  - creation of unused entries in the row directory left by deleted rows,
  - Uses an average for the number of rows per block
- The ratio is calculated for an entire segment, whereas compression is applied to individual blocks and therefore it is possible for some blocks in an extent to be compressed and others not. However, this approach is reasonable when compression is applied only to segments where data not be updated, and any new data will only be inserted except in direct path-mode.
- The formula does not take row migration or row chaining into account.
- If an object is rebuilt in parallel then each parallel process writes a separate extent. The object will have at least as many extents as the degree of parallelism. This can result in space wastage in small objects, and can even result in the object becoming larger. The formula assumes that all blocks in all extents contain data which may not be the case. These factors have the effect of reducing the block file factor calculated by the formula.

In practice, despite these limitations, the formula produces answers that are good enough for practical purposes. In practice the compressed block the file ratio is normally in the region of 400% to 500%, thus also showing the efficacy of compression, though we have observed values as low as 200%.