

DOC APAR OA56402

DFSMSdfp OAM

Publication Update Document

The z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support (Publication numbers: SC23-6866-01 and SC23-6866-30) to be updated as follows...

- Note:**
1. Red text with strikethrough should be removed.
 2. Blue text is new text that should be added.
 3. Black text is existing text for reference.

Chapter 4 -> Tuning OAM -> Tuning the DB2 databases...

Partitioning object storage tables

Because of the large amount of data that can be stored in the OAM object storage tables (the 4 KB object storage table, the 32 KB object storage table and the LOB storage structure) associated with each Object storage group, you might choose to partition the DB2 table spaces containing each of these tables. The 4 KB object storage table, the 32 KB object storage table, and LOB storage structure are each stored in separate DB2 table spaces. You might partition the DB2 table space containing any or all of these tables. [The tables can be partitioned using partition by range. In addition, the 4KB and 32KB object storage tables can also optionally be partitioned by growth \(supported as of DB2 V9\)](#)

For information about the advantages and disadvantages of partitioned table spaces, ~~see the "DB2 administration" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2)~~, please reference the DB2 for z/OS manual applicable to your DB2 version, particularly section Getting started with Db2 for z/OS -> Db2 for z/OS and SQL concepts -> Defining Db2 objects -> Creation of table spaces -> Table space types and characteristics in Db2 for z/OS.

Partitioning the 4 KB and 32KB storage structures by growth

With partition-by-growth (UTS) table spaces (supported as of DB2 V9) you can partition according to data growth, which enables segmented tables to be partitioned as they grow, without the need for key ranges.

Partition-by-growth (UTS) table spaces are universal table spaces that can hold a single table. The space in a partition-by-growth (UTS) table space is divided into separate partitions. Partition-by-growth table spaces are best used when a table is expected to exceed 64 GB and does not have a suitable partitioning key for the table.

Partition-by-growth (UTS) table spaces are like single-table DB2®-managed segmented table spaces. DB2 manages partition-by-growth table spaces and automatically adds a new partition when more space is needed to satisfy an insert. The table space begins as a single-partition table space and automatically grows, as needed, as more partitions are added to accommodate data growth. Partition-by-growth table spaces can grow up to 128 TB. The maximum size is determined by the MAXPARTITIONS and DSSIZE values that you specified and the page size.

Although a partition-by-growth table space is partitioned, it has segmented organization and segmented space management capabilities within each partition. Unlike a non-segmented structure, the segmented structure provides better space management and mass delete capabilities. The partitioning structure allows DB2 utilities to continue partition-level operations and parallelism capabilities.

For more information on partition by growth tablespaces, please reference the DB2 for z/OS manual applicable to your DB2 version, particularly section... Getting started with Db2 for z/OS -> Db2 for z/OS and SQL concepts -> Defining Db2 objects -> Creation of table spaces -> Table space types and characteristics in Db2 for z/OS -> Partition-by-growth table spaces

To create the DB2 table space containing the 4 KB or 32 KB object storage table as a partitioned by growth table space you must modify the CBRISQL0 member located within SYS1.SAMPLIB.

Note: It is possible to convert existing DB2 tablespaces to partition by growth by using the ALTER TABLESPACES command. Please reference the DB2 for z/OS manual applicable to your DB2 version, particularly section DB2 SQL -> Statements -> ALTER TABLESPACE. There are additional considerations that should be observed when dealing with tablespaces that already exist.

1. Add a new MAXPARTITIONS clause for each CREATE TABLESPACE SQL statement corresponding to the tablespace you want to be partitioned by growth. The *integer_value* value specifies an integer range that defines the maximum number of partitions to which the table space can grow.
2. Replace USING VCAT *cat_name* with USING STOGROUP *sto_group* for each CREATE TABLESPACE SQL statement corresponding to the tablespace you want to be partitioned by growth. The *sto_group* value specifies the name of the DB2 storage group that will define and manage the data sets for the table space.

Example for 4 KB and 32 KB tablespaces shown...

```
CREATE TABLESPACE OSMOTS04
  IN      osg_hlq
  USING   STOGROUP sto_group
  MAXPARTITIONS integer_value
  LOCKSIZE ANY
  CLOSE   NO
  SEGSIZE 64
  BUFFERPOOL BP2;
```

```
CREATE TABLESPACE OSMOTS32
  IN      osg_hlq
  USING   STOGROUP sto_group
  MAXPARTITIONS integer_value
  LOCKSIZE ANY
  CLOSE   NO
  SEGSIZE 64
  BUFFERPOOL BP32K;
```

For more information on the MAXPARTITIONS and USING STOGROUP clauses, please reference the DB2 for z/OS manual applicable to your DB2 version, particularly section DB2 SQL -> Statements -> CREATE TABLESPACE.

Partitioning the 4 KB object storage tables by range

During OAM installation and customization (using the default SAMPLIB members

...

Partitioning the 32 KB object storage tables [by range](#)

During OAM installation and customization (using the default SAMPLIB members

...