



Craig S. Mullins

[Return to Home Page](#)

August / September 2006



zData Perspectives

by Craig S. Mullins

[It's Time for Real Time Stats](#)

Real Time Statistics (RTS) is the first step in IBM's grand plans to bring autonomic computing to DB2 for z/OS. Introduced after the general availability of Version 7, but before Version 8, RTS provides functionality that maintains statistics about DB2 databases "on the fly," without having to run a utility program.

It seems as if there is a general wariness "out there" that is precluding widespread adoption and implementation of RTS. My very unscientific polling of DB2 user group attendees indicates that

only a smattering of DB2 shops are using RTS. May after reading this column you'll rethink your stance if yours is one of those wary shops that have yet to embrace RTS.

What are Real Time Stats?

Prior to the introduction of RTS, the only way to gather statistics about DB2 database structures was by running the RUNSTATS utility. RUNSTATS collects statistical information and stores it in the DB2 Catalog. RTS, on the other hand, runs in the background and automatically updates statistics in two special tables as DB2 data is modified. RUNSTATS is hands-on, whereas RTS is hands-off.

Now don't misunderstand, the RTS statistics do not replace RUNSTATS. Although several of the statistics are similar, RTS are never used by the optimizer to determine access paths. The RTS statistics can be used by DBAs to better administer DB2 databases.

Although DB2 is always collecting RTS data, no data will be externalized until you set up the RTS database and tables. The RTS database is named DSNRTSDB and it must contain one table space (DSNRTSTS) with two tables:

- `SYSIBM.TABLESPACESTATS` – contains statistics on table spaces and table space partitions
- `SYSIBM.INDEXSPACESTATS` – contains statistics on index spaces and index space partitions

After you create the RTS database, DB2 puts it into a stopped state. When all the objects are created you must explicitly start the

database in RW mode so DB2 can begin to externalize RTS statistics.

There are numerous statistics collected by RTS: each is defined in Appendix G of the IBM DB2 Administration Guide manual. Many columns in the RTS tables show the number of times an operation was performed between the last time a particular utility was run and when the RTS stats are written. For example, STATSINSERT in TABLESPACESTATS indicates the number of records or LOBs that have been inserted after the last RUNSTATS utility was run on the table space or partition. It is important that you initialize the RTS statistics columns. Most of these columns are nullable and if you fail to initialize the columns, the RTS values will not be correct. There are two ways to initialize these columns.

1. For each object for which you want real-time statistics, run the appropriate utility (REORG, RUNSTATS, LOAD REPLACE, REBUILD INDEX, or COPY) to establish a base value from which the delta value can be calculated.
2. Or you can create an SQL script to gather baseline statistics from the last the DB2 Catalog (as set by the last RUNSTATS) and set the appropriate RTS values.

You also can control the interval for when DB2 writes real-time statistics. The default interval is 30 minutes. To update the interval, modify the DSNZPARM named STATSINT. Of course, there are other circumstances that cause the externalization of the RTS statistics including running RUNSTATS, stopping the RTS database, and stopping DB2 MODE(QUIESCE).

Using the Real Time Stats

By implementing RTS we'll have more information at our disposal describing our DB2 objects. How should we use it? There are a lot of possible uses, but let's look at a few.

RTS statistics can help determine when to REORG. Consider this query that returns a list of table spaces with more than 100K changes, more than 25 extents, or more than 50 far indirect references:

```
SELECT  DBNAME, NAME, PARTITION, SPACE, EXTENTS,
        REORGLASTTIME, REORGINSERTS, REORGDELETES,
REORGUPDATES,
        REORGINSERTS+REORGDELETES+REORGUPDATES AS
TOTAL_CHANGES,
        REORGDISORGL0B, REORGUNCLUSTINS,
REORGMASSDELETE,
        REORGNEARINDREF, REORGFARINDREF
FROM    SYSIBM.TABLESPACESTATS
WHERE   TOT_CHANGES > 100000
OR      REORGFARINDREF > 50
OR      EXTENTS > 25
ORDER BY DBNAME, NAME, PARTITION;
```

Of course, you can change the WHERE clause to specify whatever thresholds you deem pertinent. One way to get creative with your queries is to build formulas into them. For example, the following query will return only those table spaces having more than 10% of their rows as near or far indirect references:

```
SELECT  DBNAME, NAME, PARTITION, SPACE, EXTENTS
FROM    SYSIBM.TABLESPACESTATS
WHERE   (((REORGNEARINDREF +
REORGFARINDREF)*100)/TOTALROWS) > 10
ORDER BY DBNAME, NAME, PARTITION;
```

Summary

We have only skimmed the surface of the many useful statistics offered by Real Time Statistics. If you haven't deployed RTS in your shop yet, the time is now to get them up and running.

From [zJournal](#), Aug / Sept 2006

.

© 2006 Craig S. Mullins, All rights reserved.

[Home](#).