



Craig S. Mullins

April / May 2008

[Return to Home Page](#)



zData Perspectives

by Craig S. Mullins

[Use Real Time Statistics to Automate Your Database Maintenance](#)

Real Time Statistics (RTS) have been available for several years now, but they are still only nominally implemented. Given all of the benefits that can be accrued by using RTS, it is flabbergasting that they have not been more widely embraced by DB2 administrators.

With RTS, DB2 gathers performance and maintenance statistics about database objects "on the fly," without having to run a utility program. DBAs are accustomed to scheduling the RUNSTATS utility to gather database statistics. You can think of RTS as similar to RUNSTATS, but without having to worry about when and how to run it.

Real time statistics cannot completely replace RUNSTATS because RTS are never used by the optimizer to determine access paths. But the RTS statistics can be used by DBAs to determine when to administer and maintain their DB2 databases.

If you think about this for a moment, RTS can help you to minimize RUNSTATS CPU consumption. You use RUNSTATS for two reasons:

1. To gather statistics used by the optimizer
2. To gather statistics to be used by the DBA.

After enabling RTS, you can eliminate any RUNSTATS jobs that are being run simply to obtain statistics that help you determine when to reorganize your databases. Thereafter, RUNSTATS needs to be run only for SQL optimization purposes. Indeed, by specifying UPDATE ACCESSPATH to direct RUNSTATS to collect only optimization statistics you need never collect any other statistics using RUNSTATS once RTS has been implemented.

Some shops may be able to dramatically reduce the number of RUNSTATS jobs required. If you rarely, or never, REBIND your plans and packages, and you do not use dynamic SQL in production, then RUNSTATS will rarely need to be run. Of course, I am not advocating the avoidance of rebinding your applications – in truth, I would recommend just the opposite! I am merely pointing out an opportunity to save CPU by minimizing RUNSTATS executions if you utilize this admittedly suboptimal approach to rebinding.

An additional benefit of RTS over RUNSTATS is that they are more up-to-date. Heck, the phrase “real time” is right in their name. That means the queries you run against the RTS tables will be more accurate than those you run against the standard DB2 catalog tables, which will be only as recent as your last RUNSTATS. So your decision criteria will improve when they are based on RTS instead of RUNSTATS statistics.

Optimally, you would use one of the products on the market that automate maintenance tasks based on RTS, but with some work you could “roll your own” by writing some code that feeds a scheduler. Of course, the cost of a product can be offset by the savings and accuracy it offers; most DBAs have enough to do without worrying about maintaining additional code in a “roll your own” solution.

Keep in mind, too, that DB2 9 for z/OS brings improvements to RTS. First of all, the RTS tables are moved to the DB2 catalog. Previously, users were responsible for creating and managing these tables outside the scope of the DB2 catalog. Additionally, we get some new statistics in DB2 9, the most important of which offers usage information for indexes.

Let’s talk about that in a little more detail. The LASTUSED column in the SYSINDEXSPACESTATS table contains a date indicating the last time this index was used. Any time the index is used to satisfy a SELECT, FETCH, searched UPDATE, searched DELETE, or to enforce a referential constraint, the date is updated.

A recurring historical problem in DB2 has been determining whether or not an index is being used. You can always query your PLAN_TABLEs for static SQL, but what about dynamic? That is more difficult. Now, you can simply query the LASTUSED column to see when the index was last used. Of course, you will have to give it some time -- because you might have an index supporting a rarely used query. Most shops have queries and programs that run quarterly, or even annually, but nevertheless are very important. So don’t just start dropping indexes a month after you’ve migrated to DB2 9!

In DB2 9, RTS also takes advantage of the new data types, so RTS will better document table space and index storage. The TOTALENTRIES column (for indexes) and SPACE, TOTALROWS, DATASIZE, and UNCOMPRESSED_DATASIZE columns (for table spaces) all use the BIGINT data type, which can handle values as large as 9,223,372,036,854,775,807. That's nine quintillion, if you're counting.

However, the biggest advantage of all proffered by RTS is the same as it was on day one -- the ability to use accurate, up-to-date statistics about DB2 objects to make administration decisions. The newer features have only improved our ability to do so. Continuing to ignore RTS will only make your administration efforts more difficult... and less accurate!

From [zJournal](#), Apr / May 2008

.

© 2008 Craig S. Mullins, All rights reserved.

[Home](#).