The Resource for Users of IBM zSeries & S/390 Systems
z/JOURNAL

February / March 2012

zData Perspectives

DB2 Tips

by Craig S. Mullins

In this month's column I will offer up a couple of DB2 for z/OS tips that rely on recent features that might not have caught your attention.

First up, let's talk about a recurring problem programmers and IT folks in general: and that is trying to determine whether or not a particular program is still required. As your organization grows and the number of programs increases, keeping track of them all can become quite difficult. As administration and management burdens increase, a common desire is to get rid of programs that are no longer being used. But it can be difficult to determine which programs are no longer used.

You can always "ask around," but few IT professionals would be willing to delete anything based on such a general tactic. Another common method is to review performance reports or extracts from a performance warehouse. But perhaps your

as an indicator. But DB2 10 for z/OS enhances and extends the concept of an indicator variable so they can be used outside the scope of nullability.

Consider the scenario where a program is being written to modify data. There are multiple combinations of columns that may need to be modified based on conditional programmatic processing. Maybe the program is for editing customer data. The customer has multiple columns that could be modified: name, address, telephone number, credit rating, etc. So the programmer codes up an online screen (e.g. CICS) with all of the data that can then be changed by the end user.

But what happens when the end user cracks the enter key to submit the changes? What actually changed and what stayed the same? Does the program check every value on the screen (perhaps hundreds) and build every UPDATE statement iteration for data that might have been changed? Unlikely, since that would require x! statements (where x is the total performance traces are not turned on all the time.

The question is probably more common in DB2 environments because of the plans and packages that consume storage and "sit around" taking up space if their associated program is no longer being used.

Well, for DB2 professionals this type of question becomes easier to answer once you migrate to DB2 10 for z/OS. DB2 maintains a new column, LASTUSED, in the DB2 Catalog. The column exists in both SYSIBM.SYSPACKAGE and SYSIBM.SYSPLAN and is defined as a DATE data type. The date is changed when the package header is requested from EDM. The column is also maintained for triggers and stored procedures.

Of course, you will have to give it some time -- because you might have a program that is used only rarely, yet still used. Most shops have queries and programs that run quarterly, or even annually, but nevertheless are very important. So don't just start freeing packages a month after you've migrated to DB2 10!

Another recent feature that has flown somewhat under the radar is extended indicator variables, which can be helpful to those of you who write on-line programs.

Anyone who has written a program that had to deal with possibly null results should know what an indicator variable is. Basically, DB2 represents null in a special variable known

number of columns).

Although, there are CICS options to help the programmer determine which values have changed (or simply save and compare), dealing with all the potential SQL statements could be problematic. DB2 10 indicator variables can be used to tell whether the value for an associated host variable has been supplied or not... and to specify how DB2 should handle the missing value.

This is an extended indicator variable. And it can be applied to host variables and parameter markers. Whether you will use extended indicator variables can be enabled at the package level, by using the EXTENDEDINDICATOR option of the BIND PACKAGE command; or on a statement level for dynamic SQL by using the WITH EXTENDED INDICATORS attribute on the PREPARE statement.

To utilize an extended indicator variable set its value to tell DB2 how to proceed. The values -5 and -7 can be used where -5 indicates that the DEFAULT value is to be used for the target column for this host variable and -7 indicates that the UNASSIGNED value is to be used for the target column for this host variable (in other words, treat it as if it were not specified in this statement).

With extended indicator variables then, there is no need for the application to re-send a column's current value, or to know a column's DEFAULT value. Which should make things easier for developers.

From zJournal, February / March 2012.

DB2PORTAL.com