What Does $RECPGM Do and Why Should We Use It?

March 29, 2004

The $RECPGM parameter of the SmartProduction PARMLIB (GLOBAL) member is not well understood. We will try to make its purpose and benefits clear to you here.

A quick recap from the *Administrator Guide*: "This statement defines user-developed utilities and shareware utilities used at your site to process data set records. Such utilities are used to:

- Copy all or selected records of a data set to another data set;
- Rearrange or remove record fields during copy;
- Count the number of records; and
- Show duplicate or non-duplicate records."

We are speaking about locally used programs, which are not vendor-provided utilities. Sometimes these programs have been developed at the site, sometimes they have been received as some kind of contribution from another site, from the SHARE organization, etc. If you have defined a vendor-provided utility, then please remove that name. The vendor product probably needs it. You should not replace it unless you are certain of what you are doing. Check with your vendor if you are not sure.

The $RECPGM defined programs are used by cases J240, J241 and J242.

Case J241 and J242 (cases addressing multiple passes over data) certainly apply to both efficient and non-efficient programs, so that this should not cause any problem or doubt.

Now, about J240: we do not argue that a user-written program cannot be efficient, but we are sure that the Sort product is still much more efficient. Sort is more efficient because it uses the EXCP access method, it exploits special disk and disk controller features, it sometimes fixes pages to enhance performance, and there may be other benefits based on how you have configured Sort. One has to remember that the Sort developers have had over 30 years to implement all possible tuning options.

Top efficiency for reading and writing data requires, firstly, the use of the EXCP access method. It is very complex to write code using this access method. In addition, by the way, it can only be implemented within an Assembler program. Accessing files via BSAM or QSAM, with multiple buffers (e.g., 100), even when performing I/Os in parallel (this option exists for BSAM only) - is still much less efficient than using the EXCP access method efficiently. Programs using EXCP usually read one track per I/O, sometimes even more (DFDSS: up to 1 cylinder per I/O); the savings are dramatic, in terms of number of I/Os, elapsed time, and CPU time.

In addition, Sort provides some other features, which are often not provided in a user-developed program - and when required cause local jobs to execute additional steps. For example: Sort has the ability to copy different record types from input to multiple output files, in one pass; it has the ability to reformat the input data while copying it to an output file; etc.

#