IDUG

GLOBE

January 1993

Volume 5, Number 1

How DB2 V2.3 Impacts EDM Pool Size

Craig S. Mullins works in the Technical Advisory group of PLATINUM technology, inc. *His book,* DB2 Developer's Guide, *published by Prentice Hall Computer Publishing in late 1992, contains over 1100 pages of DB2 tips, tricks and techniques (updated for V2.3). You can contact Craig via Compuserve [70410,237] or Prodigy [WHNX44A].*

DB2 V2.3 provides quite a few new application programming options; not only packages and versions, but also new program preparation options and programming features. How will these new features impact the usage and sizing of the EDM Pool? Will you need to adjust the size of the EDM Pool prior to implementing packages at your shop?

Before answering these questions, let's review the purpose of the EDM Pool. EDM is an acronym which stands for Environmental Descriptor Manager. The EDM is a component of DB2 that manages the internal structures needed by DB2 to execute application programs. It places these internal structures into an area of storage called the EDM Pool. EDM Pool storage is obtained at DB2 start-up time.

The following structures are stored in the EDM Pool for application programs to use as they execute:

- DBDs, or database descriptors. Every DB2 database has a corresponding DBD which can be thought of as an internal road map that DB2 uses to access the objects defined in that database.
- SKCTs and CTs, short for skeleton cursor tables and cursor tables. These are the executable form of your application plans.
- SKPTs and PTs, short for skeleton package tables and package tables. These are the executable form of packages.

EDM Pool

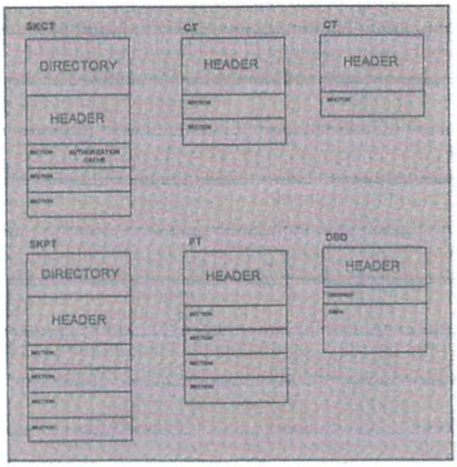

Figure 1

Refer to the diagram in Figure 1 for clarification. CTs and PTs are composed of sections. A section is made up of the SQL statement and its associated access path control structures. The EDM will read CTs and PTs from the DB2 Directory into the EDM Pool section by section. Sections are read into the EDM Pool as an executing program issues SQL requests.

The first time that an SQL statement is issued, the EDM will build a CT or PT in the EDM Pool depending upon whether the statement comes from a plan or a package. As the EDM builds the CT or PT it will also build an SKCT or SKPT. The SKCT and SKPT can be thought of as master CTs and PTs. Subsequent executions of the same plan or package will read the CT or PT from the SKCT or SKPT, thereby avoiding costly I/O to the DB2 Directory.

Each user of a plan will have their own CT in the EDM Pool and each user of a package will have their own PT in the EDM Pool. However, only one SKCT per plan or one SKPT per package will be in the EDM Pool at any one time.

Given this information, how will packages impact EDM storage? The answer is: It depends. Based upon how you implement packages and what types of plans they are replacing, packages can either increase or decrease the amount of EDM Pool storage used by an application. In general, however, after converting to packages, EDM storage needs will tend to **increase**. PTs read into the EDM Pool must be placed on a 4K page boundary,

see **How V2.3 Impacts...** *page 5*

How V2.3 Impacts...

continued from front page

whereas DBRMs bound directly into a plan need not. This can increase the storage required for package execution. Additionally, each package will have a PT header, which will consume some extra EDM storage; DBRMs do not have headers.

Also, there is one specific instance where converting from DBRMs to packages will decrease the EDM Pool storage required. If a plan with multiple DBRMs is bound with ACQUIRE(ALLOCATE), converting the plan to use packages instead of DBRMs can decrease the EDM storage needed for the plan. Packages are always executed as ACQUIRE(USE), which requires less EDM storage because certain internal structures will not be stored in the EDM Pool, but in working storage instead.

In addition to package considerations, DB2 V2.3 provides two other features which may impact EDM Pool storage usage:

- cursor with hold; and
- user specifiable authorization cache size.

When a cursor is declared using the WITH HOLD option, issuing a COMMIT will not lose cursor positioning. The benefit being that repositioning logic need not be executed after each COMMIT thereby making programs more efficient and easier to read. The cost is that a COMMIT will not free locks held on the current page (for all cursors defined WITH HOLD) nor will if free CT and PT sections from the EDM Pool. This can cause the EDM Pool to be more active as CT and PT sections "hang around" longer.

Finally, additional storage may be required for authorization caching. DB2 V2.2 automatically provided an authorization cache of 1024 bytes for each application plan. With DB2 V2.3, you can explicitly specify the size of the authorization cache at BIND time (note: there is no authorization cache for packages).

The size of the authorization cache can be from 0 to 4096 bytes in 256 byte increments. DB2 will allocate a portion of the skeleton cursor table, equal to the cache size specified at BIND time, to hold a list of authorized userids that can execute the plan. This feature reduces the amount of DB2 Catalog reading required at run time. Depending upon the nature of your applications, EDM storage may:

- remain the same if you specify CACHESIZE of 1024 for all of your plans; if CACHESIZE is not specified at BIND time, 1024 is the default;
- increase if you consistently specify CACHESIZE larger than 1K (possibly to support applications with a large user base);
- decrease if you consistently specify CACHESIZE less than 1K (possibly because of heavy PUBLIC authorization).

Overall, DB2 V2.3 will provide a more effective platform for application development. It will, however, require analyzing the EDM storage impact of the new application development features that DB2 V2.3 provides.

Improvements to IDUG Electronic Bulletin Board Being Considered

Ways to improve IDUG's Electronic Bulletin Board System (BBS) are being reviewed in an effort to make the system more user-friendly for current users and easier to access for its potential users.

Until these methods have been identified, the BBS will be off-line. Current users have been notified and IDUG will continue to keep members up-to-date on the situation's status.

IDUG Membership Benefits

Industry Contacts
IDUG provides direct access to the largest, most innovative group of DB2 users internationally and the opportunity to network and exchange information with one's peers in the DB2 community.

Voting Privileges
All IDUG user members have full voting privileges, allowing opportunities to be a part of shaping the course of this growing and vital organization.

Professional Growth
Members are welcomed and encouraged to get involved in IDUG by participating on the board of directors or on special committees.

Exclusive Vendor Participation
Vendor members attending IDUG conferences enjoy exclusive privileges, including opportunities to exhibit, host hospitality suites, conference rooms or reception rooms, and make vendor-sponsored presentations. And, of course, IDUG vendor members enjoy unique access to all user members.

IDUG GLOBE Newsletter
Only members regularly receive IDUG's quarterly newsletter, which features contributed articles by industry experts and updates on IDUG's many activities and conferences.

Innovative Programs
IDUG's annual conferences provide opportunities to share experiences and ideas with the DB2 community and to learn the latest DB2 trends and innovations.

Special Publications
Members of IDUG have access to invaluable conference proceedings publications, highlighting timely and innovative conference presentations.

Vendor Recognition Logo
Vendor members have the privilege of displaying the recognized IDUG vendor logo on company literature.