++++ SIGIMAGE ++++
2001 Ballot Extended Enhancement Descriptions --- Vers: 28 Aug 2000

Following are detailed extended descriptions for SEVEN of the 41 proposed enhancements to Image/SQL that are summarized on the SIGIMAGE 2001 Enhancement Ballot: Item numbers 04, 15, 18, 19, 35, 36, and 41.

See 2001 Ballot for HP estimate of effort required to implement each item. If you have questions about any of the 2001 enhancements or on the overall ballot process, please contact Image/SQL Advisory Committee (ISAC) by posting e-mail to: **hp3000-L@raven.utc.edu**

Note that one additional new item (# 24) has been added to the 2001 Ballot so far; since the SIGIMAGE meeting and Enhancement Ballot review at SIG3000 on 16 Feb 2000 in Sunnyvale, CA (other previously existing items "shifted down" by one; compared to the SIGIMAGE 2000 Ballot)**:**

"Add new DBINFO modes (probably '5xx') to return all root file flag settings (several flags are missing from current mode-list). Avoid need for child process & parse of DBUTIL-FLAGS output."

General Enhancements to TurboIMAGE: Extended descriptions:

(04) DBUTIL: Non-destructive creator access across groups:

Currently DBAs that have a large number of databases in different groups in the same MPE account have to go through a lot of unnecessary gyrations to do routine, non-destructive DBUTIL maintenance operations; i.e.: They have to CHGROUP to the group where the first database resides, then get in to DBUTIL, do the desired maintenance, exit DBUTIL, CHGROUP to the next database, and so on; for as many databases as they have in different groups. Some sites have 50 or more databases in 50 or more groups; they get **real** tired of multiple repetitions of this sequence. Since a DBA can do anything he or she wants to a database anyway, the current restriction does not really provide any additional security; it just makes for more work.

(15) Ability to set maximum chain lengths, with a unique key option:

This would probably involve adding a "max chain length" (MCL) command to DBSCHEMA, adding a MCL flag to the root file, and adding a new MCL error status code to DBPUT and DBUPDATE. Then both DBPUT and DBUPDATE (for CIUPDATE case) would have to be modified to check the chain length before adding or updating an entry: If chain length is already equal to MCL then set new error status and quit. Addition of the MCL parameter with ability to set MCL =1 (unique key option) for detail datasets would bring IMAGE closer to Codd's minimal definition of a relational database.

(18) Ability to pass DBOPEN to a son process, if database is "stable":

The ability for a program to pass a DBOPEN to another program or a son process would be very useful, and would be expected in many cases to allow avoidance of unnecessary duplicate DBOPEN sequences. However, as a practical matter making this work reliably while the database is in the process of being actively updated and / or while locks are in place is an extremely difficult if not impossible proposition. But it is likely this would still be very useful even if the feature was restricted to times when the database was in a "stable state"; i.e.: All pending transactions completed, and all locks released. HP has indicated that with these qualifiers, it should be a lot easier to implement the feature.

(19) IMAGE Linear Fast MASTER Dataset Expansion / "Inflation":

Currently there are two ways to expand the capacity of a TurboIMAGE MASTER Dataset: Rehash with the Tool of your choice (or DBUNLOAD / recreate / DBLOAD, if anybody still does that); and relying on the Dynamic Dataset EXpansion (DDX) for MASTERs feature. Because of the nature of hashing, the locality of the old entries may not be preserved, resulting in the new entries being distributed in varying ways; i.e.: for large MASTERs the cost in time to rehash can be large. Depending on DDX for MASTERs guarantees that chain lengths will keep growing. This proposal is to offer a third option for quickly increasing the capacity of MASTERs without DDX:

Every master would have a scale factor; initially = 1. Note that "deflation" (scale factor < 1) would NOT be allowed. IMAGE hash value computation would not change; but the primary entry location would change to be:

SIGIMAGE 2001 Ballot Extended Descriptions
Vers: 28 August 2000

hash_value times scale_factor (rounded down). Benefits of this method would be twofold:

[a] It would reduce the "third bear of IMAGE" problem, because "ventilation holes" would be added, making DBPUT faster (until they fill up, of course).

[b] It requires only a single sequential read of the old master, and a single sequential write of the new master, with no other (disk) work. It is therefore expected to be much faster than rehashing.

More details at: **http://www.allegro.com/sigimage/inflate.html**

Enhancements that may benefit from the IMAGE ENCHILADA:

ENCHILADA = (ENhancement for caCHIng Limited Authorized DAta).

More details at: **http://www.allegro.com/sigimage/index.html**

(35) Image/SQL support for Binary Large Objects (BLOBS).

A BLOB may be anything from a short text string or a signature to a fully illustrated chapter from a technical manual or a full-length feature movie. Original suggestion was to store all BLOBS in auxiliary storage, not in-line in IMAGE itself (as with ALLBASE and most other RDBMS). IMAGE would only keep pointers to the BLOB. This would keep IMAGE databases small enough to be backed up, but retain necessary flexibility to evolve with changing needs without modifying the database structure. Follow-on idea was that it should be possible to store relatively small BLOBs directly in IMAGE; in which case max size of an "in-line" IMAGE BLOB would probably be bounded by the MPE Transaction Manager (XM) max limit.

(36) Allow positive data in PACKED / ZONED decimal Items to be defined in IMAGE as UNSIGNED (P/Z) or SIGNED (P+/Z+):

The ISAC did not agree what to do if a program tries to put an UNSIGNED value in a "P+" field; or a SIGNED positive value in a "P" field. Some thought such data should be rejected; others felt that it should automatically be converted into the proper P/Z or P+/Z+ format (assuming of course that it was valid P/Z data in all other respects). If HP agrees to implement this enhancement, further discussion of this issue is in order.

(41) An ENCHILADA "Tag" for "Linked Procedures", (might facilitate approximation of RDBMS Stored Procedures in IMAGE):

This request is to allow creation of an ENCHILADA link between IMAGE data items and fields and an NMPRG file or NM XL. If DBUTIL was then enhanced to ENABLE a particular database for DATAVALIDATION, the idea would be for any addition or modification of data associated with that item or field to be processed by that NMPRG or XL. Data validation errors would be reported back to the program that attempted to add or modify the data. Note in this scenario failure (however that was defined) would **only** result in the return of an error; no IMAGE data would be changed.

A second potential scenario is for an additional modification to DBUTIL, to allow ENABLE dbname FOR DATAMODIFICATION. In this case, in addition to (perhaps) optionally returning an error message of some kind, the "linked" NMPRG or XL would **automatically** correct incoming data. A prime example of where this might be used is for data items that have been defined as UNSIGNED positive P or Z. If a program attempted to DBPUT or DBUPDATE an item defined as UNsigned with a SIGNED positive value, the linked NMPRG or XL would change it to UNSIGNED on the fly.

This proposal is controversial, especially the 2nd scenario where IMAGE would gain an "active" capability to manage and modify data. ISAC member Fred White has pointed out that IMAGE was designed to be a dataBASE management system, NOT a DATA management system. Whether the ability to improve and implement "single point of control" for business rules would be a convincing and valid reason to expand IMAGE capabilities in this area may be debated for some time. In any case, an ENCHILADA would only provide the **vehicle** for any such subsequent implementation; it would not by itself cause anything to happen….