



# Newsletter

The SIGIMAGE Newsletter is published by the INTEREX Special Interest Group for IMAGE/SQL Databases (INTEREX is the International Association of Hewlett-Packard Computer Users)

Volume 7.1 — March 1998

## Table of contents

|  |    |
|--|----|
| <i>From the Chair</i> .....  | 1  |
| <i>Hobson's Choice</i> .....   | 4  |
| <i>Transaction logging</i> .....   | 4  |
| <i>Enable TurboIMAGE databases for dumping</i> .....   | 5  |
| <i>Editorial contact</i> .....   | 5  |
| <i>The Coming Storm</i> .....  | 5  |
| <i>DDX problem &amp; solution</i> .....  | 7  |
| <i>Robelle's support of Hourglass, SetDate, and Time Machine</i> ..                              | 9  |
| <i>The Developer's HP 3000 918/DX</i> .  | 9  |
| <i>OLR for 7x24 operations</i> .....   | 10 |
| <i>A QUERY update, 1997-12-05</i> . . .  | 11 |
| <i>Copyright information</i> .....   | 11 |
| <i>SIGIMAGE mission statement</i> . . .  | 11 |
| <i>Guidelines for reblocking data sets</i>   | 12 |
| <i>1997 SIGIMAGE Enhancement Ballot</i> .....  | 13 |
| <i>1998 Enhancement Descriptions</i> .   | 14 |
| <i>SIGIMAGE Enhancement Requests Now / Soon Available in Image/SQL As of February 1998</i> ..... | 19 |
| <i>Are you a member of SIGIMAGE?</i>   | 20 |

## From the Chair

*Ken Sletten*  
*SIGIMAGE Chairman*  
*NUWC Division Keyport*  
*Keyport, WA, USA*

## The Year in Perspective

1997 was another active and productive year for SIGIMAGE, that saw the delivery of a number of new enhancements to Image/SQL and TurboIMAGE. Two of these new features deserve special mention: The availability (as of MPE/iX 5.5 Express 3) of TurboIMAGE intrinsic B-Tree indices (a.k.a. sorted sequential access), and the bundling of a 16/32-bit ODBC driver with Image/SQL (ODBCLink/SE).

The additional flexibility and performance provided at no additional cost by the bundling of these two features makes them a major milestone for IMAGE. Readers who want to learn more about B-Trees and ODBCLink/SE should refer to the *Communicator 3000* for MPE/iX-5.5-Express 3 (Powerpatch C.55.03): A detailed technical review of B-Tree indices (and other enhancements) is on pages 4-9 through 4-18 of that *Communicator*; information on ODBCLink/SE can be found on pages 2-10 through 2-18.

B-Trees and ODBCLink/SE were not the only new features delivered

in 1997. At least seven additional IMAGE enhancements hit the street:

- Make CIUPDATE = ALLOWED the default.
- Increase XM rollback limit (2 MBytes to 4 Mbytes).
- DBINFO Mode 406 (fully qualified database name).
- QUERY support of IMAGE intrinsic B-Trees.
- QUERY NOMATCH option.
- Increase dataset max entry length to 2378 Words.
- Multiple Dependency Semaphores (DSEM) scalability.

Along with all the new features that were released in 1997, one temporary delay needs to be reported: Because of all the other work recently in process, Master Dataset Dynamic Expansion (MDX) slipped and did not make the Express 4 release of 5.5. The SIEC expects that MDX will be available in the near future.

A total of 11 enhancement requests were moved to the "Now/Soon Available in Image/SQL" list for 1998 (the complete "Now/Soon" list can be found on the last page of the "SIGIMAGE 1998 Enhancement Descriptions" that accompanies the 1998 Enhancement Ballot). These 11 items include the last six "delivered" features listed immediately above, plus the following five in-process enhancements that will be available "soon". Readers should

note that soon is a fairly “elastic” word; i.e.: implementation details and delivery schedules for most of the following items have not been published:

- QUERY to show all OMNIDEX Indexes.
- Make Native Mode QUERY the default (in 6.0).
- DBUTIL option to confirm PURGE and ERASE.
- > 4 MBytes XM rollback limit mitigation.
- IMAGE datasets > 40 GBytes.

Twenty-three items should have been on the Now/Soon list for 1997 (what are now Items # 19, # 20, and # 21 were inadvertently omitted when the 1997 version of “Now/Soon” was first published). The 11 added for 1998 means that a total of 34 individually tracked IMAGE enhancements have been moved to the “Now/Soon” list since 1990. Not too bad for a product that is going into its second quarter-century.

And in the “not too bad for SIGIMAGE” category, for the seventh year in a row in 1997 the SIG continued the tradition of holding two “face time” meetings each year: SIGIMAGE met in extended session at IPROF-97 in March, and again at the HP World conference in Chicago during the last week of August.

One item that stands out as a “left-over” from both IPROF and HP World in 1997 is the issue of possibly providing date/time datatypes in Image/SQL. Especially at IPROF-97 there was a wide range of views on how this enhancement should be implemented, if at all. It was discussed there at some length and with considerable passion, but in the end nothing even remotely resembling a consensus was reached. A few months later the same issue came up again at SIGIMAGE — Chicago. In the intervening months after IPROF, thinking on this subject appears to have “gelled” (at least for the people at the Chicago meeting): The near if not actual 100 percent consensus was that IMAGE should support the SQL date/time standard.

Date/time data types in Image/SQL was the # 4 rated enhancement after the votes were counted for the 1996 SIGIMAGE Enhancement Ballot. It was rated # 3 after the tally for the 1997 Ballot. It remains on the 1998 Ballot.

It is probably fair to say that at HP World 1997 Chicago senior HP managers said even more nice things about the HP 3000 in public than they did at HP World 1996 Anaheim. And Anaheim was considered a good year. The biggest single headline was probably the public commitment by HP-CSY General Manager Harry Sterling to “64-bit MPE”. The details on just what “64-bit MPE” will end up meaning for the 3000 in general and IMAGE in particular is something that users can look forward to hearing more about in 1998.

## ***Expansion of IMAGE Internal Limits***

The last item on the 1998 “Now/Soon Available” list is another feature that clearly falls into the “special mention” category: Datasets larger than 40 gigabytes. It will involve the expansion of at least one fundamental internal limit that has been in place since IMAGE was first invented in the 1970’s: The maximum number of blocks in one IMAGE dataset.

The current EntryByName scheme maxes out at  $2^{23} - 1 = 8,388,607$  blocks. Since BLOCKMAX is 2560 half words (5120 bytes), the maximum size of one IMAGE Detail dataset is currently just over 40 Gbytes using JUMBOS. Given that the record pointer limit is 255 entries per block, this obviously also limits the maximum number of entries in a dataset. If a site has individual records large enough that they must use Block Factor = 1, then 8,388,607 records per dataset is a hard upper limit. As the number of records in high-end TurboIMAGE datasets continues to grow and accelerate, breaking through this limit from the earliest days of IMAGE is becoming more and more important for the largest HP 3000 installations.

HP is still working out the implementation details for datasets > 40 GB, so it is not possible at this time to say what the new upper limit for dataset size will turn out to be. However, discussions between the SIGIMAGE Executive Committee (the SIEC) and HP lead the writer to expect that the increase will be a large number, in both absolute and percentage terms. More information on the details of this in-process enhancement are expected to be available later in 1998. It is certainly expected to be one of the discussion topics at the SIGIMAGE meeting at IPROF-98 in March.

## ***The SIEC Gets Even Larger***

The SIEC (SIGIMAGE Executive Committee) continues to serve as the on-going liaison between IMAGE users and the HP R&D Database Lab. In 1996 we had a total of 19 people including the Chair on the SIEC, with six of these 19 members being HP employees. By the end of '97 the SIEC had grown to 29 people, of which 10 work for HP. The size of this Executive Committee and the depth of Image technical knowledge represented by its far-flung members continue to be unique among Interex SIGs.

*Current members that are not HP employees:*

Neil Armstrong, Wirt Atmar, Denys Beauchemin, Gary Biggs, Steve Cooper, Jerry Fochtman, Joe Geiser, Chris Hagood, Jeanette Nutsford, Terry O’Brien, Ken Paul, Alfredo Rego, Tom Renz, Gilles Schipper, Stan Sieler, Ken Sletten, Rich Trapp, Fred White, Rene Woc.

*HP members of the SIEC:*

Jon Bale, Subba Rao Bheema, Dianna Carter, Tien-You Chen, Bharati Desai, Vikram Kumar, Shobha Pradeep, G Ramamoorthy, Kriss Rant, V S Subrahman-  
yam.

## ***The 1998 SIGIMAGE Enhancement Ballot***

The HP R&D Database Lab continues to be funded and staffed, with a portion of that funding allocated for additional work on enhancements to TurboIMAGE and Image/SQL. Two of the key factors HP uses to determine which enhancements will be done next are the results of the yearly SIGIMAGE Enhancement Ballot and the follow-on Interex System Improvement Ballot (SIB). The items relating to IMAGE on the SIB (that is sent out to all Interex members) are those that get the most votes on the latest SIGIMAGE Ballot. Therefore results of this ballot will set the context for enhancement priority feedback to HP for all IMAGE users.

I bring the above to the reader's attention just as I and other Chairs have done in years past, in the hope that I can encourage every IMAGE user around the world that reads this issue of the Newsletter to take the next important step:

### ***PLEASE SUBMIT A 1998 SIGIMAGE ENHANCEMENT BALLOT***

Excuse the shouting, but this is a classic case of "every vote counts". If you do not give HP your priorities for IMAGE enhancements, others will make the decision.

All readers also please note that there are some departures from prior years regarding both the format and the submission of the 1998 Ballot:

- The initial release of the 1998 Ballot was distributed to users that attended the SIGIMAGE meeting at HP World in August 1997. This "Version # 1" Ballot had 61 uncommitted items still outstanding. Since that Chicago meeting, six of those 61 enhancement requests have been moved to the "Now/Soon Available in Image/SQL" list; plus one important new request (# 53.2) on PACKED/ZONED decimal data items has been added (the SIEC talked about this one for months).
- Because 10 percent of the items on the initial version of the 1998 Ballot have now been moved to the "Now/Soon" list, before sending the ballot out to the entire user base it was decided to publish a revised "Version # 2" Ballot. Version # 2 is what is enclosed with this Newsletter.
- Votes on a "Version # 1" Ballot will still be counted, *unless* users that sent them optionally choose to also

submit a Version # 2 Ballot, in which case the Version # 2 entry will replace and supersede their prior vote. Especially if you voted earlier for items that have subsequently moved to "Now/Soon", you are encouraged to submit a Version # 2 Ballot.

- Another reason for submitting a Version # 2 Ballot is that HP and the SIEC agreed that it would be useful to try and get user feedback as to *why* users are asking for particular items. Therefore five "Reasons" columns have been added to the Ballot in Version # 2. See Ballot sheet for detailed instructions on filling it out.
- Unless you are very familiar with the details and ramifications of each item, before voting readers are advised and encouraged to review the extensive additional information for most enhancement requests that is provided in the "1998 Enhancement Descriptions – Version # 2" that accompanies the Ballot. The wording of each enhancement on the Ballot sheet itself is very abbreviated.
- Voting on your 1998 Ballot by electronic means is strongly encouraged. A URL to allow voting via browser is being set up by a member of the SIEC. This URL was not available as of press time for this issue of the Newsletter, but it is expected that it will be posted to the HP3000-L Internet discussion group soon. It should also be printed on hardcopy versions of the 1998 Ballot that accompany this Newsletter and that will be handed out at IPROF-98.
- For those who might be unable to submit a ballot via the web but have e-mail, an e-mail version can be obtained by contacting the SIGIMAGE Chairman at: [sigimage@interex.org](mailto:sigimage@interex.org) (but for ease of electronic tabulation, please vote via browser if you can).
- IMAGE users who attend IPROF-98 can fill out and submit their 1998 Ballot before they leave the conference.
- Or you can FAX a hardcopy Ballot to Interex at: 408-747-0947. If all else fails, you can—as in prior years—send in your Ballot by snail-mail to Interex at the address given on the Ballot.
- Regardless of means of submission, Ballots must be received by Friday 3 April 1998 to help determine which enhancement requests go forward to the 1998 Interex SIB. Ballots received after 3 April but before HP World – San Diego will still be added to the final 1998 SIGIMAGE vote totals that are forwarded to HP, but they will not influence the Interex SIB results.

## ***Coming Attractions***

SIGIMAGE is scheduled to have its next "face time" meeting at IPROF-98 in March. The SIG will also meet for two hours at HP World – San Diego in August 1998.

For info on both IPROF-98 and HP World, visit <http://www.interex.org/index.html>

## ***Acknowledgments***

Continued from last year: All my fellow members of the SIEC, both users and HP employees, for their contributions of time and effort. And IMAGE users around the world, for their primary contribution to the continued and expanded use of the product and the HP 3000.

---

## ***Hobson's Choice***

*Chris Hagood*

*FHS*

*Pueblo, CO, USA*

Several recent discussions have prompted me to ponder why seemingly reasonable enhancements to Image seem so hard to get. I thought that I would share my thoughts with my fellow Image-a-philes.

In regard to enhancements, it seems that one of Image's biggest strengths is causing one of its biggest problems. In my opinion, one of Image's biggest strengths is its wide and varied use. There are many mission critical applications based on the speed and stability of TurboImage and there are many fine tools that have been developed to enhance the use of TurboImage. However, this broad user-base constitutes a tremendous inertia that must be overcome whenever Image is to be changed. HP is under much more pressure to maintain backward compatibility than many of its competitors because of this user-base. In addition, the effort required to test any change must be huge and the probability that something, somewhere will stop working because of a change is great simply because of all the places that Image is used.

On the other hand, HP's competition had the benefit of the years of experience that users had building applications using previous database management systems (including Image) before they even entered the marketplace. This experience allowed database users to expand and refine their requirements far beyond what was

imaginable when Image was designed. Some of these requirements include: databases that are self-describing, databases that can tune themselves, databases that can be dynamically redefined, databases that are aware of the data they contain. The knowledge of these new requirements combined with the explosive growth in the power of computers allowed the competition to develop database management systems that are very appealing to database users today.

Unfortunately, to incorporate these new requirements into Image requires significant change and, as noted above, the inertia of a large user-base tends to increase the effort required to make these changes. So we are faced with Hobson's Choice: let Image fall further behind what the majority of database developers expect or enhance Image aggressively and accept the costs and risks associated with that enhancement. Perhaps there is a third choice. Build a new Image that takes the best of TurboImage and incorporates the best of the new requirements and provide a migration path from TurboImage to this new Image for those willing to walk with Image into the next millennium.

---

## ***Transaction logging***

*Gilles Schipper*

*GSA Inc.*

*Thornhill, Ontario, Canada*

On a Wednesday afternoon about six months ago, I answered the phone. The voice at the other end of line sounded very nervous.

He identified himself as a contract programmer for a company who was/is a customer of mine. He also mentioned that Joe Jones (not his real name), the system manager of this large HP3000 shop, had suggested to him that he call me. Joe Jones was the person I normally dealt with at this company.

I soon understood why he was so nervous. Moments earlier, he tried to PURGE a data base using MPEX, thinking he was logged on to the TEST account.

By now you've correctly guessed that he was actually logged on as MANAGER.PROD,DATA and not as he had thought, MANAGER.TEST,DATA, and he had managed to actually delete all the database's unopened data sets.

Fortunately for him, this HP3000 installation was utilizing IMAGE transaction logging.

Knowing that, I assured the poor guy at the other end of the line that I would have his data base recovered in about an hour without skipping a beat.

An hour later, the contract programmer breathed a huge sigh of relief, 120 or so HP3000 users were back on-line after a brief period of inconvenience, and IMAGE transaction logging saved the day (and lots of money) for this long-time HP3000 customer.

This is a true story which illustrates how advantageous transaction logging can be — in some cases even more valuable than mirroring or raid discs, which I hasten to add can also serve as important data protection mechanisms.

By the way, after reading this war story, some readers may ask, why not just restore the purged datasets from backup? After all, these datasets were not open while the attempt to purge the database was made and therefore were probably not modified since the last backup.

We had actually thought of that, but then dismissed this option after considering that indeed those “quiet” datasets would indeed have been modified by some of the many batch jobs that were executed in the time between the previous full backup and the Wednesday of the “almost disaster”.

Space” or “freaddir error 0 on root file” occurs, the DBG will be disabled. This will require all users to get out of the database before anyone can get back in again. On the other hand, if dumping is not enabled and you get this error, then DBPUT just returns a -3, the DBG is not affected, and users will *not* have to get out of the database.

---

## **Editorial contact**

*F. Alfredo Rego*  
*Adager Corporation*  
*The Adager Way*  
*Sun Valley, ID 83353-3000 — U.S.A.*

*Tel. +1 (208) 726-9100 — Fax +1 (208) 726-8191*  
[alfredo@adager.com](mailto:alfredo@adager.com)  
<http://www.adager.com>

---

## **Enable TurboIMAGE databases for dumping**

*Ken Sletten*  
*NUWC Division Keyport*  
*Keyport, WA, USA*

One feature of the DBUTIL *enable* command that is often overlooked is the *dumping* parameter. If a database ever has a problem that results in a TurboIMAGE abort, with *dumping enabled* the system will generate two special privileged mode files (file code = -403). The naming convention of these files includes a leading “I” or “J” followed by seven digits giving the Julian day of the year and the time. These files can be very helpful to the HP Response Center in trying to figure out why an abort has occurred.

Dumping only comes into play in the rare case of a TurboIMAGE abort, and does not cause any performance penalty during normal operations.

However, users should be aware that there is a least one case where the end result with dumping *enabled* is quite different: If dumping is enabled and a “Lost Free

---

## **The Coming Storm**

*Wirt Atmar*  
*AICS Research, Inc.*  
*University Park, NM, USA*

There is currently a severe problem in IMAGE that will only become more apparent as more people attempt to utilize it in a client/server environment, especially when using ODBC access. The problem is one of definitional ambiguity — and it’s always been there. It’s just that it hasn’t been much of a problem to date, so it’s gone relatively unnoticed. IMAGE currently allows P (packed) and Z (zoned) dataitems to contain either signed or unsigned data in the fields, without type checking or any reference information as to what the correct format should be.

Packed and Zoned (P/Z) datafields are not like integers. In an integer field, you only have two possibilities for the format of a number: positive or negative. But in P/Z datafields you have three cases: negative, positive unsigned (the way an integer number is represented), or

positive signed (a “+” sign always accompanies the number).

The rub that now occurs exists because many PC-based programs expect to see only signed P/Z data, although the vast majority of the P/Z data in current IMAGE databases is unsigned. Trying to mix the two formats is very much like trying to shift into reverse at 80 miles per hour. What you get when you try is a minor catastrophe.

Joe Geiser recently explained the basic problem to the SIGIMAGE Executive Committee (SIEC):

Zoned Decimal and Packed fields, when written from an application using IMAGE/SQL (such as Visual Basic or Visual C++), will always populate the field with a signed number. Not a problem in itself, except that IMAGE supports UNSIGNED positives. In the project which revealed this, a COBOL program blew sky high when it read data written to a data item defined as Z4 in Image, 9(4) in the COBOL program (note the absence of the “S”), and had “094B” in the field.

To the VB application, this was 942. To Image, it was just a bunch of bits, to the COBOL program, it was an invalid numeric. This is the cause of the problem. We want IMAGE/SQL to be able to write “0942” instead of “094B” —hence the change to IMAGESQL, where the root problem lies.

A proposal has been made to solve the immediate problem by modifying the IMAGESQL UPDATE TYPE command to translate a signed numeric into unsigned by default (or, alternatively, to allow you to specify the format as signed or unsigned). Unfortunately, this approach is not a true solution. Indeed, it’s not much more than a temporary fix.

Modifying SQL alone will not solve the core problem, especially if the intention is to access pre-existing data that was (and perhaps continues to be) put into the database by a wide variety of tools, simply because it does not address the definitional ambiguity that lies at the heart of the current confusion.

A far better method is to explicitly state the type of data that is expected to exist in the data fields by creating two new dataitem types in addition to the existing P and Z datatypes: P+ and Z+. The P+/Z+ notation would explicitly mark the datafield as anticipating that all numerics entered into that field would be signed — while the P/Z notation would expect only unsigned positive numbers.

The intention behind this P/P+/Z/Z+ proposal is to create a situation that will:

- Have totally consistent data (all unsigned or all signed values) within every record of a dataset, dataitem by dataitem.
- Provide a mechanism so that corrupted dataitems (containing part signed, part unsigned values in the same dataset) can be cleaned up by third-party database tools.
- Be fully compatible with all prior usage (no host-based application breaks because NOTHING changes).
- Perform autotranslation of the dataitem value when necessary to satisfy non-HP3000, PC-based database accessing programs.
- Be transparent to all processes and all users.

Under this P/P+/Z/Z+ proposal, no existing compiler would need to be altered. Nor would QUERY or any existing application program need to be modified. In most cases the proposal would be completely painless to existing users. The only programs that would need to be modified would be: (i) third-party database tools (but only so that they may better support the new, more explicit datatypes), (ii) programs like Suprtool and QueryCalc that bypass the IMAGE intrinsics to retrieve data by directly accessing IMAGE datafiles, and (iii) programs like SQL and ODBC, but only for their own purposes of representing signed data in their preferred manner.

Three IMAGE intrinsics would need to be modified to implement this proposal: DBPUT, DBGET, and DBUPDATE. The manner by which these intrinsics would operate would be as follows.

DBPUT/DBUPDATE would work this way:

- If the root file datatype was P, and the value given to the intrinsic was a legal, unsigned bit pattern, then the data would be placed in the database unmodified. If on the other hand the value given to the intrinsic was a legal, signed bit pattern, it would be converted automatically to an unsigned bit pattern, a trivial process.
- If the root file datatype was P+, and the value was a legal signed bit pattern, then the data would be placed in the database unmodified. If it were a legal, unsigned bit pattern, it would be converted to a signed bit pattern.

Automatic sign conversion would be a new feature in IMAGE. In the past, if someone wished to write “CATSUP” into a datafield of any type, it was allowed. IMAGE has traditionally allowed such corruption and



there may be good reason to continue the tradition (although such trickery makes life very difficult for generations of users and should be strongly eschewed). The other choice is to now and in the future reject such entries, which, in itself, would be simple enough if IMAGE were already checking for legal patterns.

In either circumstance, with or without strict integrity checking, only legal patterns would be autotranslated as to sign.

DBGET would work in an analogous manner:

- If the root file datatype was P and the data retrieved was a legal, unsigned bit pattern, then the data would be presented unmodified. If on the other hand the retrieved data was a legal, signed bit pattern, it would automatically be converted to the unsigned format. If the root file-listed datatype was P+, the reverse would apply.
- If the retrieved data was not a legal pattern, it would be presented as is, regardless of root file datatype or stringent data value checking.

This way, if current P/Z usage in the database is unsigned (which is the great majority of current usage), NOTHING would need be modified.

On the other hand, if P/Z items were being used implicitly by an application as signed items, those dataitems' definitions would need to be changed to P+/Z+ in the IMAGE root file, either by schema modification and a DBUNLOAD – DBLOAD cycle or through the use of a third-party database tool.

But that's it for traditional, HP3000-based programs. Data would still be presented to all applications as in the past. Data extraction programs that download information to an external source whose wish it is to see P/Z data as exclusively signed or unsigned would also need to be modified — but that is true in either case.

Definitional ambiguity is the bugaboo that lies at the core of the current problem. With explicit data typing, that confusion completely disappears. Both ends of the equation are now known. An external program would know how the data is formatted and can convert it at will to whatever format it desires.

It's important to note that under this proposal, the IMAGESQL change (UPDATE TYPE as SIGNED or UNSIGNED) would be unnecessary. SQL and ODBC could present legal bit patterns in either signed or unsigned format and IMAGE would autotranslate the data into the format specified by the root file — and thereby absolutely guarantee acceptability to all other current HP3000 usage. Only on extraction would the SQL/ODBC class of programs need to monitor the

source data format and perform any necessary conversions.

Making this change now is important, before it becomes an extremely critical item (as it is almost certain to become). Perhaps even more important yet is to do the modification correctly — and only have to do it once.

---

## ***DDX problem & solution***

*Jon Bale  
Hewlett-Packard Company  
Cupertino, California, USA  
Bangalore, Karnataka, India*

Date: Mon, 3 Nov 1997  
From: Jon Bale <[jon\\_bale@HP.COM](mailto:jon_bale@HP.COM)>  
Subject: DDX Problem

Since there was some discussion in October on this list about a problem with DDX, I thought I'd share what we at HP now know about this issue.

HP has received reports of a problem being experienced by seven IMAGE/SQL users who have databases containing one or more data sets using the Dynamic Detail Dataset Expansion (DDX) feature. The reported problem relates to data stored in a data set which has undergone some dynamic expansion—that is, it has “filled up” and its capacity has been automatically increased. The problem encountered by these users is that some data which is subsequently written to the data set is placed “beyond” the end-of-file (EOF), and once the database is closed and reopened, that data is inaccessible. If a program attempts to read that data or add more data entries in the same area, it gets an error -212, “Database is Corrupt.” The Service Request documenting this is SR #5003-367607.

Working with some of our independent software tools vendors, we have located a defect in the TurboIMAGE code which would cause the observed effect. In the case we found, an erroneous expansion may take place, which can then lead, possibly at a much later time, to data set entries being placed beyond the real file EOF. When the file is later closed, these entries are lost. The circumstances in which the erroneous expansion could take place are as follows: At the same time that an actual dynamic data set expansion is being performed within a

call to DBPUT by one process, a second process (user) accessing the same data set for the first time causes execution of IMAGE's "open data set" module. If the timing is just right (actually just wrong), the result will be incorrect internal data set structural information being saved to disk. Specifically, the recorded internal CAPACITY has been incremented TWICE, while the EOF has been correctly incremented—only once. This defect has been in the code since the introduction of DDX approximately three years ago.

We have developed a fix for TurboIMAGE that removes this defect. Patches will be created for MPE/iX 5.0 and MPE/iX 5.5. The MPE/iX 5.0 patch is TurboIMAGE version C.06.23 and has Patch ID TIXKX11. The patch for MPE/iX 5.5 is TurboIMAGE version C.07.07, has Patch ID TIXKX13, and is built on the version of IMAGE released on MPE/iX 5.5 Express 3.

Customers who use DDX and think they may be susceptible to the problem described (see below) can request either of these beta patches from the HP Response Center.

### ***How can you tell if you already have this problem?***

If you do not use Dynamic Detail Dataset Expansion, you are not affected. If you are a heavy user of DDX with frequent data set expansions, you might be in the "high risk group." The defect we have identified can only affect detail data sets that are enabled for DDX on which an expansion actually takes place. If the erroneous combination of operations occurs, the result would be a data set whose logical EOF (current capacity) exceeds the corresponding file's physical EOF.

You can check for this condition using the FORM SETS command of QUERY, which gives the Current Capacity (CC) and Blocking Factor (BF) of each data set. Calculate the MPE EOF of a DDX data set using the formula,  $(CC + (BF - 1)) / BF$ . Verify this calculated EOF with the EOF given by the MPE command :LISTF DDXSETnn,2 where "DDXSETnn" is the file name of that DDX data set. If the two EOFs do not match, there is a problem.

If you have and use one of the IMAGE/SQL structure maintenance tools provided by HP or an independent vendor, that tool may also have the capability to locate instances of this problem. If you discover that one of your data sets is so afflicted, it may be possible to correct the problem using the same tool. (See your tool's documentation.)

### ***How can you avoid experiencing this problem on your image database?***

Although we cannot be sure that the defect we repaired is the unique cause of this problem, I will assume that it is, for the purpose of addressing this question. The simplest answer, then, is: Install and use one of the TurboIMAGE beta patches mentioned above. This is the only option that I recommend.

However, if you are unable to acquire and install the patch immediately, consider the following alternative: Until you install the patch, avoid opening a data set from one process while another is expanding it! This may be easier said than done, but here are some ideas, each one independent of the others.

- Access the database exclusively.
- Open all DDX-enabled data sets needed from each process before allowing any process to add data to any of these sets.
- Preallocate and pre-expand DDX data sets by adding in exclusive mode many new dummy entries (more than your actual processing will) and then deleting them before your real processing begins.
- Disable DDX on any nearly-full data set, using an HP or third party tool in which you have confidence.

There are certainly variations on the ideas mentioned above, and perhaps better ones.

My belief and hope is that only a few customers have actually been affected by this particular defect. However, our goal is for no user to lose data, and data loss IS a possibility in this instance. Since we now have identified the likely cause and have a solution, I thought it was worth sharing with everyone.

---

### ***On 18 January 1998, Jon Bale wrote:***

This is an update to my November 3 post on the DDX problem. In that post, I said, "We have developed a fix for TurboIMAGE that removes this [DDX] defect. Patches will be created for MPE/iX 5.0 and MPE/iX 5.5. The MPE/iX 5.0 patch is TurboIMAGE version C.06.23 and has Patch ID TIXKX11. The patch for MPE/iX 5.5 is TurboIMAGE version C.07.07, has Patch ID TIXKX13, and is built on the version of IMAGE released on MPE/iX 5.5 Express 3. Customers who use DDX and think they may be susceptible to the problem



described... can request either of these beta patches from the HP Response Center.”

### **The news:**

- The patch built for MPE/iX 5.0 — TurboIMAGE C.06.23, patch ID TIXKX11A — is being put into GR (general release) status this week.
- TurboIMAGE C.07.07, patch ID “family” TIXKX13, was found to be defective, although no customer who installed the patch encountered the problem. The replacement for C.07.07 is TurboIMAGE C.07.09. (Customers who had been given the C.07.07 patches have been notified that they should take and use C.07.09 instead.) The TurboIMAGE C.07.09 patches are now available, in beta status. They are TIXKX45A for MPE/iX 5.0 and TIXKX45B for MPE/iX 5.5.

Special note for MPE/iX 5.0 users: Such customers would only choose to install TurboIMAGE C.07.09 (TIXKX45A) if they want the DDX fix AND the new features shipped to 5.5 users on Express 3 (B-Trees, ODBCLink/SE, etc.). In that case, they also need to install the “G2” versions of IMAGE/SQL and ALL-BASE/SQL, which contain related changes. However, in other cases, these customers can get the DDX fix (without the new features) in TurboIMAGE C.06.23 (TIXKX11A), now in general release.

## **Robelle’s support of Hourglass, SetDate, and Time Machine**

*Robelle Tech Support* ([support@robelle.com](mailto:support@robelle.com))  
Canada and Anguilla

If you are using SetDate, Time Machine, or Hourglass to help you with your Year 2000 MPE testing, then read on.

This year three new pieces of software were created for the HP 3000 as a result of users needing to test application programs for year 2000 compatibility. These software tools allow you to set the date in a localized manner, for a program or a session, without changing the system date or rebooting the system.

The three products are *HourGlass 2000* from Allegro Consultants, *Time Machine* from SolutionSoft Systems, and *SETDATE* from Hewlett-Packard.

Robelle products employ very tight integrity checking which verifies, among other things, that the system date appears to be correct. Our products ran fine with *HourGlass 2000*, but would not run with *SETDATE* or *Time Machine*.

Robelle software has been changed to be more tolerant of *SETDATE* and *Time Machine*’s method of setting the system clock. This change is available starting in the following versions of Robelle software: *Qedit 4.6.02*, *Suprtool 4.0.13*, *Xpress 3.2.01*, *Spell 1.7*, *Xpedit 1.6*, *SmartDate 2.1*, *HowMessy 2.4.01*, *Select 3.8*.

*Qedit 4.6.02* is the newest production release which is being distributed right now to all *Qedit* users on support. *Suprtool 4.0.13* is available on demand as a pre-release version to any *Suprtool* user on support.

## **The Developer’s HP 3000 918/DX**

*Jeff Vance*  
*Hewlett-Packard Company*  
*Cupertino, California, USA*

An HP 3000 918/DX is a generously priced bundle of hardware and software exclusively available to HP 3000 developers.

Here’s the configuration for the MPE 918/DX Developers Platform:

- 918/LX with 64 MB memory
- 4 GB low-profile fixed disk drive
- 4 GB DDS DAT drive
- integrated 802.3 LAN
- single-ended SCSI 2
- 600VA UPS
- system console
- pre-loaded with: MPE/iX OS, POSIX Developers Kit, ARPA Services, documentation on CD.
- 8 user license, but we plan to offer a 4 user license and we have priced this 918 at what the 4 user license price will be. This should help get lower s/w pricing from other vendors.

## ***Included from HP***

### ***HP Languages***

- C/iX
- Pascal/iX
- Fortran 77/iX
- COBOL II/iX
- Business Basic/iX
- Transact/iX
- RPG/iX

### ***HP software***

- Image/SQL
- Allbase/SQL
- Glance/iX
- AIFs
- XDB

***There is also a wealth of discounted and/or free software from Third Parties.***

For more details regarding the HP 3000 918/DX, visit <http://jazz.external.hp.com/papers/918dx/faq.html>

---

## ***OLR for 7x24 operations***

*Denys P. Beauchemin  
Hi-Comp America  
Houston, Texas, USA*

The past few years have seen many enhancements to IMAGE, which make it almost possible to envisage having a 7x24 environment. The DDX, MDX, Jumbo Datasets and On-Line backups are the components I am talking about. However, there is still a component missing. I am referring to On-Line Reorganization of detail datasets or OLR.

Nowadays one can backup the database, have it expand when more entries are needed and have the datasets grow to staggering sizes. Over time however, the detail datasets get fragmented and the chains lose their efficiency. At some point, the application can slow down to the point it is impacting overall performance, as it chases down fragmented chains across a large dataset.

As databases get bigger, it becomes more difficult to find a time to do that most daunting of tasks, a detail

dataset reorganization. Currently, the database must be accessed exclusively by the utility performing the dataset reorganization. While this reorganization is going on, no other application can use the database. So even though the computer is running, the system is effectively down for the users.

There is a way to do an on-line reorganization, safely and simply. It uses some of the technology which has been made available over the last 7 years for IMAGE. Namely, High water mark DBPUT and dynamic transactions. The principle is as follows: at some point, the dataset to be reorganized is switched to *hwmpu*t so that all subsequent entries added will go to the end of the dataset and push up the high water mark. The entries vacated below the high water mark will not be reused until the *hwmpu*t is disabled. Once this is done, the OLR then starts reading the primary master, or whatever the user wants to use as the "organizing master".

Now we want to husband our resources here. There is no need to reorganize every chain. We want to reorganize the chains that would benefit the most from a reorganization. This means chains that are in multiple fragments and chains that go one way in the dataset, then the other way and so on.

An algorithm could be designed to highlight the chains needing reorganization. When such a chain is encountered, the OLR begins a dynamic transaction and locks the dataset along the path. The OLR then starts down the chain, DBGETs an entry, records the position in the dataset and DBPUTs the entry, effectively at the end of the dataset along with any new entries added by other programs running concurrently. Then the OLR repositions itself on the entry it read and DBDELETES it, and DBGETs the next entry on the chain. A simple loop which has the effect of following a chain made up of multiple fragments and positions it in one contiguous segment at the end of the dataset. If any problems are encountered during the move of the chain, the OLR simply calls DBUNDO or aborts. Once the chain is finished, the transaction is committed and the locks are released. The OLR reads the next master record and decides whether it needs reorganization or not.

The process is running in user mode, using IMAGE intrinsics. If the process aborts, or is aborted, IMAGE will rollback the current transaction, (in this case the chain being reorganized) and will leave the database intact. If the system fails during the process, the database will be recovered automatically at startup time, and the transaction in progress at the time of failure will be rolled back. In any case, the work performed up to that point will be preserved and the database will be intact.

The OLR could be set to be interrupted, either by aborting the job or by setting a system flag such as cre-

ating an empty file, which informs the OLR to stop processing. This flag could be checked at the end of every chain.

The OLR could also be instructed to work between 10:00PM and, say, 4:00AM. Or it could be instructed to read 10% of the master dataset, or other combinations. If the OLR finishes gracefully, it could record in a small control file the address or value of the last master record checked/reorganized. This way, it could be restarted from that point on. If a restart from the beginning is desired, one would simply purge the control file.

The OLR is not meant to replace a regular detpack or reorganization, but rather to extend the time between these vital, albeit time-consuming tasks. Also, an OLR would contribute to making the regular detpacks and reorganizations much faster by virtue of the fact it would have ordered many of the records in the detail dataset. A properly written detpack or reorganization function will take advantage of any existing order.

If it were possible to reorder the delete chain while the database is active, then it would be possible to disable the hwmput and reuse the entries below the HWM. Even if all the free entries are not contiguous, at least the delete chain would start at the beginning and move forward and not jump all over the dataset.

And SR D500-336727 VERSION command shows NOT available for program versions, has been fixed.

The focus of any work on QUERY at the moment is to improve the situation of retrievals involving TPI indexes. There are many problems with complex retrievals ignoring the TPI items, SUBSET with TPI selection parameters, etc. We plan on working with the vendors to try to improve the integration.

Alternatively, Query may be changed to reject TPI retrievals that it can't handle properly. Obviously, the former is preferred, but the latter is a possible option to avoid customer problems.

The FIND FIELD=FILEDB issue is "still under investigation".

---

## ***A QUERY update, 1997-12-05***

*James Overman  
Hewlett-Packard Company, SSG  
Roseville, California, USA*

For MPE 6.0, QUERY.PUB.SYS will be a copy of QUE-RYNM.PUB.SYS. For those requiring the old CM version, that will be in QUERYCM.PUB.SYS.

The FO INDEXES command has been tweaked to display all TPI index items (even those with the same name as a straight IMAGE item). Also, the dataset name for the item is now displayed. NOTE: SHOW INDEX is different than FORM INDEXES. SHOW INDEX gives the Identity of the TPI package being used. This change should be retrofitted to Query for MPE 5.5 and 5.0 on the next patch in early 1998 in Version D/N.03.12 or later.

Also, SR 5003-254037 TPI find w/unsigned or + key on ZONED/PACKED items returns each value twice, has been fixed.

---

## ***Copyright information***

You are welcome to reproduce and distribute the articles that appear in The SIGIMAGE Newsletter.

Please give credit to the authors and to The SIGIMAGE Newsletter, a free publication which INTEREX provides as a courtesy to SIGIMAGE.

---

## ***SIGIMAGE mission statement***

SIGIMAGE's goal is to provide a forum for fostering mutual help and cooperation among its members.

SIGIMAGE represents the interests of its members to Hewlett-Packard.

SIGIMAGE is dedicated to working with HP in furthering the capabilities of IMAGE, for the continuing benefit of all IMAGE users throughout the world.

## Guidelines for reblocking data sets

Tom Renz  
COTC Computer Consulting  
Pueblo, CO, USA

Reblocking TurboIMAGE data sets is a commonly overlooked option available to today's DataBase Administrator (DBA). If implemented early the disc space savings could prevent the purchase of additional disc drives sooner than anticipated or if implemented too late it could take a large number of hours to reblock in order to recover the wasted space. Some guidelines can be applied when making a determination on whether to reblock a data set or an entire database. Tools are avail-

able that can assist you with your reblocking activities including *Adager*, from Adager Corporation, and *DB General*, from Bradmark Technologies, Inc. All illustrations and examples were gathered from a recent run of *Adager* for purposes of this discussion.

Several key items are available after a request for a database reblock has been made. These items include the current blocking factor, media length and disc utilization, and the potential blocking factor, media length, disc utilization and disc space savings for the requested maximum media length. As a general rule use the maximum media length of 2048 words as your new block size since MPE reads data from disc into memory in 4096 word chunks. Refer to Table 1 for an example of the current and recommended figures.

**Table 1:**

|                |   | Current Block ----- |      |       |         | Recommended Block ----- |      |       |                     | Saved<br>sectors |
|----------------|---|---------------------|------|-------|---------|-------------------------|------|-------|---------------------|------------------|
|                |   | Fac                 | Lgth | Usage | Sectors | Fac                     | Lgth | Usage | Sectors             |                  |
| AUTO-SET-1-A   | A | 29                  | 640  | 100%  | 235568  | 87                      | 1920 | 100%  | 235568              |                  |
| AUTO-SET-2-B   | A | 60                  | 1024 | 100%  | 1344    | 120                     | 2048 | 100%  | 1360                | -16              |
| AUTO-SET-3-C   | A | 60                  | 1024 | 100%  | 83408   | 120                     | 2048 | 100%  | 83408               |                  |
| AUTO-SET-4-D   | A | 15                  | 1006 | 98%   | 1066688 | 19                      | 1275 | 99%   | 1052656             | 14032            |
| MANUAL-SET-1-A | M | 18                  | 1010 | 98%   | 4464    | 25                      | 1402 | 99%   | 4416                | 48               |
| MANUAL-SET-2-B | M | 13                  | 768  | 100%  | 787808  | 26                      | 1536 | 100%  | 787808              |                  |
| MANUAL-SET-3-C | M | 16                  | 881  | 98%   | 4592    | 37                      | 2038 | 99%   | 4544                | 48               |
| MANUAL-SET-4-D | M | 11                  | 628  | 98%   | 8752    | 29                      | 1655 | 99%   | 8640                | 112              |
| DETAIL-SET-1-A | D | 4                   | 617  | 96%   | 2500016 | 9                       | 1387 | 98%   | 2444464             | 55552            |
| DETAIL-SET-2-B | D | 3                   | 853  | 95%   | 73200   | 4                       | 1137 | 98%   | 70592               | 2608             |
| DETAIL-SET-3-C | D | 7                   | 988  | 96%   | 1911072 | 9                       | 1270 | 99%   | 1857984             | 53088            |
| DETAIL-SET-4-D | D | 4                   | 897  | 87%   | 4000016 | 9                       | 2017 | 98%   | 3555584             | 444432           |
| DETAIL-SET-5-E | D | 5                   | 896  | 100%  | 1211152 | 11                      | 1970 | 96%   | 1258352             | -47200           |
|                |   |                     |      |       |         |                         |      |       | Total sectors saved | 522704           |

Using the above table, review each column to determine whether to reblock a data set or not. Each column is evenly weighted when making a decision, but there are times when a "gut" decision overrides the initial choice. When making your decision, you can reblock all of the data sets, selectively reblock specific data sets, or leave them as they are.

Time is another important factor that should be considered when making the decision on which data sets to reblock. Once the data sets have been reblocked, no additional time in the future will be required. If caught early, the time requirements will be minimal, but as data sets grow, the time necessary to reblock them may exceed a weekend.

Starting with data set "AUTO-SET-1-A" you can see that the *Current Blocking Factor* is 29 and the *Recommended Blocking Factor* is 87. The disc utilization remains at 100% and there is no disc space savings. This data set should be reblocked in order to utilize the increased number of records per block. This increased number provides an increased chance that a record is in memory when data access is requested as well as a greater probability to place synonym chain entries in the current memory block when entries are added.

Moving on to data set "AUTO-SET-2-B" the *Current and Recommended* entries are the same except the number of sectors saved would be decreased by 16 sectors. This data set is not a large data set and could be reblocked quickly, if you choose. In order to keep a consistent blocking factor for all data sets it is recommended that this data set be reblocked.

For data set "AUTO-SET-3-C" the column entries are the same as "AUTO-SET-2-B" except that no sectors will be saved. Since this data set is still relatively small it is recommended to reblock this data set to the recommended size. The reblock function may take approximately five minutes to transform but remember that you will need to reblock only once.

For data set "AUTO-SET-4-D" the *Current and Recommended* columns change by small increments but the sector savings exceed 14,000 sectors (approximately 3.6MB). Improved efficiency of disc space (99% versus 98%) and more records per block (19 versus 15) are also key factors on basing your decision. Based on all of these factors it would be recommended to reblock this data set.

All of the remaining data sets except "DETAIL-SET-5-E" can be scheduled for reblocking due to overall disc space savings, percentage of disc usage and increased records per block. Data set "DETAIL-SET-5-E" should be skipped due to reduced disc usage and loss of disc space, even though the number of records per block more than doubled. If skipping the "DETAIL-SET-5-E" data set from reblocking and choosing all of the other data sets, a savings of 569,904 sectors (approximately 145.9MB) will be gained.

By reviewing all of the data sets that can be transformed via a reblocking operation and making decisions based on the basic numbers, a more efficient database can be realized based on:

- Disc usage.
- Lowered disc space requirements.
- Increasing the number of records per block.

Reblocking your database will not impact your programs due to structural changes because the changes are at the file system level. The file system will present the

individual records to your application in the same manner as before, one record at a time.

Privileged third-party applications that use multi-record-nobuf access will not be affected, because they dynamically increase/decrease their buffers based on the file layout and definition.

What you will gain is more throughput from the third-party tools (when they access your data block-by-block or multiblock-by-multiblock) as well as file system and memory management efficiencies.

Reblocking a database is a one-time effort for existing databases and should be addressed when new databases are created.

Who knows? You may save a byte or two.

---

## ***1997 SIGIMAGE Enhancement Ballot***

### ***Final Results + HP Response to Leading Requests***

*Ken Sletten*  
*SIGIMAGE Chair*  
*NUWC Division Keyport*  
*Keyport, WA*

The final "Top 20" ranking from the 1997 SIGIMAGE Ballot is given below. A total of 74 valid ballots were received and tabulated (nine more than were returned in 1996). All 74 ballots properly cast 100 votes.

Bundled ODBC access direct to TurboIMAGE without the need to go through Allbase was the overwhelming winner. Bundled ODBC access direct to MPE & KSAM files without Allbase was ranked number two. Readers might note that the number one SIGIMAGE enhancement request from 1996 was "Provide a bundled 32-bit ODBC Driver for Image/SQL". This 1996 request was delivered to the user base in 1997 in the form of ODBC Link/SE, which does not provide an option to bypass Allbase.

Given the recent delivery of bundled ODBC Link/SE, HP consistently made it clear throughout 1997 that CSY has no plans to provide either of the top two 1997 requests for direct ODBC access as bundled features; main reason being that both of these features have been available for some time from third-party vendors. Prior to HP World Chicago I posted a question asking for clarification on this issue at the Management Round-



table. The official HP answer recently appeared on the Interex web site:

*“Our position on this enhancement remains unchanged from IPROF97. Our policy is to promote the use of third-party products that complement HP’s overall offering. This allows us to focus our resources on core technologies that only HP can provide, such as IMAGE. By working with our partners, rather than competing with them, HP is able to offer much broader and richer solutions to our customers.”*

Image/SQL users are of course completely free to keep asking HP to bundle direct ODBC access to TurboIMAGE and MPE & KSAM files. Both of these requests will remain on the 1998 SIGIMAGE Ballot as item # 1 and item # 2. However, the above statement in combination with other comments made by various HP managers at IPROF97 and HP World Chicago seem to constitute a pretty clear Company policy that CSY is not going to bundle ODBC direct access to TurboIMAGE or MPE & KSAM files any time soon, if ever. Therefore if these features are important to your site, I would suggest contacting one of the third-party vendors that have products that already provide this capability. I tend to believe HP when they say bundling is not in the cards.

In the interest of conserving space in this edition of the Newsletter, the following “Top 20” 1997 enhancements (out of 48 total) are presented in compact format, listed in highest to lowest votes order. The first number (“# dd”) is the enhancement rank. Second number (“Edd”) is the item number on the 1997 Ballot. The 3rd number (“Vddd”) is total votes for each enhancement. The 4th number (“Bdd”) is the number of ballots that gave that enhancement > 0 votes. Last is a short description of each enhancement. Note that # 08 (E16) is now available as FREEWARE and # 09 (E47) has moved to the “Now/Soon Available” list.

**# 01 E01 V1504 B44 HP bundled ODBC access direct to IMAGE, without going through Allbase.**

**# 02 E02 V0599 B31 HP bundled ODBC access direct to MPE & KSAM files, without Allbase.**

**# 03 E35 V0426 B34 Date/time datatypes.**

**# 04 E36 V0337 B22 Binary Large Object (BLOB) support in and/or through IMAGE.**

**# 05 E04 V0313 B16 Ability to add/drop datasets, indexes, and items on the fly (SQL DDL).**

**# 06 E29 V0308 B23 FIND Field A = Field B in IMAGE QUERY.**

**# 07 E03 V0233 B15 Attach MPE/KSAM to a DBE.**

**# 08 E16 V0207 B15 Be able to retain and reuse parameters across DETACH / re-ATTACH.**

**# 09 E47 V0193 B16 DBUTIL option to confirm PURGE and ERASE, displaying full DB name.**

**# 10 E38 V0177 B19 Ability to specify all DBUTIL flags and settings in DBSCHEMA.**

**# 11 E26 V0170 B10 Measurement Interface (MI).**

**# 12 E06 V0163 B10 Read-only DBLOCK Mode.**

**# 13 E05 V0159 B14 Support SQL NULL Item.**

**# 14 E37 V0158 B12 GETUPDATE w/o DBGET.**

**# 15 E48 V0151 B12 Add DBMGET to FTP/iX.**

**# 16 E15 V0147 B14 Easier access from POSIX.**

**# 17 E12 V0145 B10 ODBC access TO clients.**

**# 18 E30 V0134 B12 MULTIFIND in QUERY.**

**# 19 E27 V0130 B13 Allow associate tracking files.**

**# 20 E42 V0127 B13 Improve performance for DBUPDATE of sort fields.**

---

## 1998 Enhancement Descriptions

Ken Sletten  
SIGIMAGE Chairman  
NUWC Division Keyport  
Keyport, WA, USA

The following detailed descriptions provide additional information about many of the 1998 SIGIMAGE Enhancement Proposals for Image/SQL that are summarized in the accompanying Ballot. After expanded descriptions of the 56 items on Version # 2 of the 1998 Ballot, there is an additional list of enhancements that are already done or have been committed for delivery by HP: The “Now/Soon Available” list. Version # 1 of the 1998 Proposals had 61 items still outstanding; six of these 61 have been moved to Now/ Soon; one new

enhancement (# 53.2) on P/Z data items has been added.

These 56 Enhancement Proposals are grouped in seven categories: Those to Support Direct ODBC Access; to Support SQL Access; IMAGESQL Utility Enhancements; Vendor Needs; QUERY Enhancements; General Enhancements; and Scalability (the last is new in 1998). For year-to-year "traceability", items that were on the 1996 & 1997 ballots are numbered as before, resulting in gaps where enhancements are OBE or done (#s 24, 34, 34.1, 40, 44, 47, 58, and 59). Except for #14, items added in 1998 are in the QUERY section or at the end. Items new for 1998 are identified by a "[98]".

See ballot sheet for estimate of HP effort required to implement each item.

Directions for filling out and sending in the 1998 Ballot can also be found on the ballot itself (one-sheet, two sides). If you have questions about any of the 1998 enhancements or on the overall ballot process, please contact a member of the SIGIMAGE Executive Committee (SIEC) directly. E-mail addresses for user-members of the SIEC are on the accompanying "1998 Ballot - Additional Information" sheet. Or, you can do one of the following:

- Send e-mail to [sigimage@interex.org](mailto:sigimage@interex.org)
- Post a message to the HP3000-L discussion group (send e-mail to [hp3000-l@raven.utc.edu](mailto:hp3000-l@raven.utc.edu))
- Call Interex 800-468-3739 (fax 408-747-0947)

### **Enhancements to Support Direct ODBC Access to HP 3000 Data**

1. *Provide bundled 16/32-bit ODBC client access direct to TurboIMAGE, without having to go through Allbase/SQL.*

Some users only require access to IMAGE data from ODBC-compliant tools on clients, and do not need to be able to ATTACH an IMAGE database to a DBE on the 3000. This request is for an HP-supported ODBC TurboIMAGE direct access method. Note that this capability can be purchased now from third-party vendors.

2. *Provide bundled 16/32-bit ODBC client access direct to MPE and KSAM files, without having to go through Allbase/SQL.*

This request is for HP-supported ODBC direct access to MPE and KSAM files. As with ODBC access to TurboIMAGE, this capability is available now from third-party vendors. It is *not* currently possible to ATTACH MPE and KSAM files to an ALLBASE DBE (see number 3 below).

### **Image/SQL Enhancements to Support SQL Access:**

3. *Ability to ATTACH MPE and KSAM files to an Allbase/SQL DBE (full integration with Allbase/SQL).*

This is asking for an enhancement to the IMAGESQL utility program to allow it to handle MPE and KSAM files in essentially the same manner as IMAGE databases; or, alternatively, for a separate utility program to handle non-database files. Note that while an existing IMAGE database is self-describing, this would require that the record layout for MPE and KSAM files be defined in some manner by the DBA.

4. *Ability to add/drop Image datasets, indexes, and items without having to get all other users out of the database (SQL DDL).*

Currently, the only method of adding or dropping datasets, items, and indexes in IMAGE is by doing an UNLOAD-change-RELOAD cycle, or by using a database tool. With some restrictions, SQL-compliant RDBMS can perform these tasks online. A true "Relational IMAGE" would need to have the same capability.

5. *Support SQL NULL item functionality in Image/SQL.*
6. *Provide IMAGE read-only DBLOCK mode (repeatable read).*

This would allow multiple concurrent reads, but would force any DBLOCK for write access to be treated serially and wait for the read-only lock to be released. Some users do not implement IMAGE locking to get faster read throughput; but in so doing risk having the underlying data change.

7. *Multi-record/multi-set DBGET, DBPUT, DBUPDATE.*

This is asking for these existing intrinsics to optionally be able to act on entire blocks of IMAGE data, instead of just on individual records.

8. *Support SQL SAVEPOINT in Image/SQL.*

SQL has the concept of being able to mark a point in the current transaction which can then be rolled back to, using the SQL *rollback work* statement.

9. *Ability to better inform the SQL Query Optimizer about the most efficient retrieval path to use in accessing IMAGE data.*

10. *Allow default values for fields not specified at DBPUT time.*

This would allow fields in an IMAGE dataset to take on a user-specified default value, if another value is not expressly provided during a DBPUT.

11. *Allow one transaction to span both ALLBASE and IMAGE.*

Currently, SQL applications cannot update both IMAGE and ALLBASE in the same transaction. For sites that need to work with data stored in both types of DBMS, this makes it much harder to maintain logical integrity.

12. *Allow ODBC access TO clients FROM Image/SQL.*

Image/SQL ODBC access is currently a one-way pipe from the client to the server. This is asking for bidirectional ODBC access capability.

13. *Make DBSCHEMA fully SQL-aware.*
14. *Bundle all of Allbase/SQL with Image/SQL (users currently get only 8 out of 9 Allbase software modules). [98]*
15. *Easier access to Image and Allbase from POSIX programs.*

#### **Enhancements to the IMAGESQL Utility Program**

16. *Bundled ability to retain & reuse attach parameters across a DETACH /re-ATTACH of Image/SQL database to a DBE.*

This would provide the ability to store and re-use mapping kept in the ATCINFO file that details SPLIT, UPDATE, and ADD USER. These mappings are lost at DETACH time, and must be re-entered in some manner after a subsequent ATTACH. Providing a supported method of doing this would make administering Image/SQL much easier. Note that this capability is available now as *freeware* from a third-party vendor.

17. *Improve DBA functions for managing authorization Ids.*

The current method of adding and managing users requires IMAGE database passwords that are difficult to maintain and hard to change, especially with a large number of users. This asks for the ability to utilize user-classes when adding and managing user IDs.

18. *Provide the ability to enable/disable SQL access without having to ATTACH/DETACH an IMAGE database to/from a DBE.*

19. *Allow DBA user to ATTACH an IMAGE database without password.*

In most cases it is actually *less* secure to require a DBA to provide a maintenance word, since maintenance words end up being recorded in relatively insecure log files.

20. *Do not require exclusive DBE access for all IMAGESQL functions.*

21. *Provide an IMAGESQL option to NOT attach IMAGE Auto Masters.*

In most cases, SQL users are not interested in accessing AUTO Masters. For IMAGE databases with many datasets, this would significantly reduce the clutter when selecting tables through SQL and ODBC.

22. *Allow wild-card ADD USER in IMAGESQL.*

23. *ATTACH command to register IMAGE Master - Detail referential integrity constraints in a DBE.*

This is to make ALLBASE aware that IMAGE DETAILS must have corresponding MASTERS, so that SQL INSERT and DELETE will verify. This should be done automatically at ATTACH time.

24. *ATTACH option to load ODBCVIEWs. (\*\* DELETED in 1998 \*\*)*

This enhancement request is OBE with the release of the bundled ODBC Link/SE product in MPE/iX 5.5 Express 3. ODBC Link/SE completely replaces the PC API product that this enhancement was targeted at.

#### **TurboIMAGE Vendor Enhancement Needs:**

25. *Provide a user-accessible DBQUIESCE intrinsic.*

HP has an internal implementation of DBQUIESCE, focused on the needs of online backup products. This asks for DBQUIESCE functionality to be opened up, so that database tools and utilities can make use of this feature to access an IMAGE database without having to get all users off. This may allow some of the third-party vendors to provide features that address some of the other enhancement requests on this list.

26. *A Measurement Interface (MI) for performance data.*

This asks for something analogous to the MPE MI that would allow tools to provide detailed performance data for TurboIMAGE.

27. *Allow tracking files to be associated with an IMAGE database.*

Various tools need to be able to determine the total collection of all files that should be associated with a database. This would make it easier for tools such as backup products to reliably STORE and RESTORE all files needed for that particular database, such as the schema, third-party index files, jumbo-dataset chunks, etc.

28. *Tool "memory area" file(s) associated with an IMAGE database.*

Many tools would benefit from being able to store tool-specific data in one or more files associated with a database. This would allow this data to be included in any transfer of the database as an entity, without the need to track it as a collection of unconnected files. This would also allow logging of structural problems, which database tools could then use in attempts to correct them.

**Enhancements to TurboIMAGE QUERY:**

29. *Ability to do FIND FIELDA = FIELDB.*

This request is currently under investigation by HP - Roseville. The amount of effort to implement by the lab will depend in considerable degree on what if any limitations may be acceptable. Examples: Must the fields be compatible data types; in the same dataset; in the same database; etc.

30. *Provide a MULTIFIND update / delete function in QUERY.*

31. *Ability to access DBCONTROL and DBINFO from QUERY.*

With the advent of such features as TPI, dynamic expansion, and B-trees, users need to be able to access and control them through QUERY. HP has said that doing DBCONTROL will probably be easier than DBINFO, so this request may be split into two separate enhancement requests. Note that QUERY is non-privileged and so cannot do many DBCONTROL calls.

32. *Allow data retrieval using IMAGE record number in QUERY.*

There is at least some divergence of opinion on this one. Two members of the SIEC voted for it on the 1998 Ballot, while one member put five votes against it. Reason for vote against was that if a dataset is ever repacked, then of course IMAGE record numbers will change and that could lead to very unhappy results. If this request was implemented, users would need to be aware and remember that IMAGE record numbers are transitory.

33. *Support IMAGE dynamic transactions in QUERY.*

To maintain data integrity, QUERY needs to support DBBEGIN / DBEND and DBXBEGIN / DBXEND transactions.

34.2 *Have FIND, MULTIFIND, and SUBSET display a running count of the number of records retrieved that is updated periodically, without the need for the user to do a <CTRL-Y>. [98]*

**General Enhancements to IMAGE:**

35. *Provide Date/Time datatypes in Image/SQL.*

36. *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. Originally this suggestion was to store all BLOBS in an auxiliary storage mechanism, not in-line in the database 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 the necessary flexibility to evolve with changing needs without modifying the database structure. Subsequently there was a follow-on suggestion that it should be possible to store relatively small BLOBs directly in an IMAGE database. If this was done, the maximum size of an "in-line" IMAGE BLOB would probably be limited by the largest single transaction that could be handled by the MPE Transaction Manager (XM), which is currently 4 MBytes.

37. *GETUPDATE to update record(s) without prior DBGET.*

This asks for a new GETUPDATE intrinsic, to perform a record retrieval and update in one call. It would save the overhead of conducting parameter checking multiple times; besides, it would simplify coding for applications that just want to change the value of a field regardless of the current value.

- 38. *Ability to specify all DBUTIL flags and settings in DBSCHEMA.*
- 39. *Non-destructive creator access across groups in DBUTIL.*
- 41. *DCE compatibility for IMAGE (Security, directory services, etc.).*
- 42. *Performance improvements for DBUPDATES of Sort Fields.*

With CIUPDATE, it is now possible to use DBUPDATE to change a sort field value in place, instead of having to DBDELETE / DBPUT the entire record. This proposal is to have DBUPDATE compare the new value to the old and begin the search for the new position either forward or backward on the chain, based on the "direction" of the change in the value — instead of always starting the search for the new location at the end of the sort chain.

- 43. *Global files (avoid max files per process limit + better performance).*

The ability at FOPEN time to obtain a system-wide "handle" for a file, not just a process-local handle, which could be communicated to any other process on the system. If in privileged mode, that other process could then access the file via file system intrinsics. This feature would allow IMAGE to improve performance for second and subsequent DBOPENS, and also avoid the limitation on the number of files that can be opened by a single process (which we are approaching again with jumbo chunks and with index files).

- 45. *Have DBUTIL ignore database file equations (can't use them anyway).*
- 46. *Ability to set maximum chain lengths in IMAGE.*
- 48. *Extend FTP/iX command set to include a DBMGET that takes rootfile input and insures complete copy of an IMAGE database:*

A DBMGET in FTP would operate similarly to MGET, but would take only an IMAGE root file as a valid argument before copying an entire database.

- 49. *No more "stealth" flags in DBUTIL; allow SHOW all. [98]*
- 50. *Augment DBSCHEMA with GUI-based product that can also handle SQL management functions. [98]*

The current need to use various combinations of DBSCHEMA, IMAGESQL, ISQL, and SQLUTIL to manage Image/SQL and Allbase/SQL operations is too complex and does not provide a standard user interface.

- 51. *DBSCHEMA option to allow Masters after Details. [98]*
- 52. *DBSCHEMA option to allow symbolic expressions for capacity. [98]*
- 53. *DBSCHEMA optionally automatically figure out PATH count. [98]*
- 53.1 *Option to force a unique "primary key" in DETAILS. [98]*

This would probably involve adding a unique key command to DBSCHEMA, adding a unique key flag to the root file, and adding a new unique key error status code to DBPUT and DBUPDATE. Then DBPUT and DBUPDATE (with the CIUPDATE option) would have to be modified to check the length of the chain before adding or updating; if the length is already > 0 then set the new error status and quit. The addition of a unique key option for detail datasets would bring IMAGE closer to Codd's minimal definition of a relational database.

- 53.2 *Allow non-negative data in PACKED/ZONED (P/Z) decimal Items to be defined as either UNSIGNED (P/Z) or SIGNED (P+/Z+). [98]*

The SIEC had a prolific, extended e-mail discussion about this issue over a period of nearly two months. Prior to the advent of Image/SQL and the IMAGESQL Utility program (previously called Allbase Turbo Connect), positive P/Z data in TurboIMAGE was almost universally UNSIGNED. Negative P/Z data must of course have the sign overpunch. It turns out that positive P/Z data entered by any ODBC driver going through Allbase is always SIGNED. This means that if a site has existing non-negative data in a P/Z field and then starts to use ODBC via Allbase to add new data, they will almost certainly end up with a mix of both SIGNED and UNSIGNED positive values. While a person can immediately see that "123" and "+123" are equivalent, the same is not true for computer programs. Therefore this has the potential to create a royal mess for non-negative P/Z data.

User-members of the SIEC believe that having SIGNED as the only choice for positive P/Z data coming in to TurboIMAGE via Allbase is a defect and not an enhancement request. And while implementing the ability to define datatypes in TurboIMAGE as either P/Z or P+/Z+ is the ultimate goal, it may turn out that a fix



for the current “via Allbase” P/Z problem can be delivered faster. This would most likely be done by introducing two new options for the IMAGESQL UPDATE TYPE command, SIGNED and UNSIGNED, for non-negative data in P/Z Items. User-members of the SIEC recommend that if the UPDATE TYPE enhancement is done as an initial step to address the SIGNED / UNSIGNED P/Z issue, then the default should be UNSIGNED.

On the desired end result of being able to define P/Z and P+/Z+ Items in TurboIMAGE, one issue on which the SIEC did not universally agree was what to do if a program tries to put an UNSIGNED value in a “P+” field; or conversely, a SIGNED value in a “P” field. Some thought that 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.

#### ***Image/SQL Scalability (performance and features)***

54. *Allow Dynamic Detail Dataset Expansion (DDX) with sets > 4 GByte (DDX with JUMBO datasets). [98]*

55. *Allow more than 199 datasets per TurboIMAGE database. [98]*

Initial investigation by a user-member of the SIEC has indicated that increasing the number of datasets to right around 255 should be relatively easy. Allowing more than 256 datasets would be a much bigger task.

56. *Allow more than 1023 data items in one TurboIMAGE database. [98]*

57. *Allow more than 16 paths per dataset. [98]*

Initial investigation by a user-member of the SIEC has indicated that increasing the number of paths to right around 255 should be relatively easy from the internal tables point of view. Even so, it seems unlikely that 255 paths would ever be needed. However, in many cases a good argument can be made for more than 16 paths, especially with the advent of IMAGE intrinsic B-Trees.

60. *Support data replication with foreign databases. [98]*

## ***SIGIMAGE Enhancement Requests Now / Soon Available in Image/SQL As of February 1998***

*Ken Sletten  
SIGIMAGE Chairman  
NUWC Division Keyport  
Keyport, WA, USA*

The following list contains most enhancements that have been—or are about to be—implemented in Image/SQL. These enhancements were a direct result of users working together through SIGIMAGE as our advocacy forum to HP since 1990. Except for # 7 (coming SOON), the first 30 items are all available NOW. The last four items are in the “being worked on” queue.

Special mention should be made of the last item: “Increase internal limits to allow datasets > 40 Gbytes”. Delivery of this enhancement will involve the expansion of fundamental IMAGE internal limits that have been in place since IMAGE was invented in the 1970s. Implementation details for this feature are still being worked out, so it is not possible at this time to specify what the new upper limit for dataset size will turn out to be, or when the enhancement might be delivered. But current expectation is that the increase will be a large number, in both absolute and percentage terms.

1. *Critical Item Update (CIUPDATE) for Search Keys and Sort Items.*
2. *Bundling SQL as part of Image, instead of making it a separate product.*
3. *Increase IMAGE dataset capacity to 4 GB.*
4. *Increase IMAGE dataset capacity beyond 4 GB (currently at 40 GB).*
5. *Removal of the need for LG capability for User Logging.*
6. *Detail Dataset Dynamic Expansion (DDX).*
7. *Master Dataset Dynamic Expansion (MDX).  
<< Coming SOON >>*
8. *DBPUT and CIUPDATE with \$NULL Search fields.*

9. *Control of Free Entry List usage.*
10. *DBGET Error 18, broken chain retry.*
11. *Improved detection of database corruption during DBPUT.*
12. *Image/SQL being able to take advantage of Third-Party Indexing (TPI).*
13. *Item level locking in Image/SQL.*
14. *Native Mode QUERY.*
15. *Sorted Sequential Access (a.k.a. IMAGE intrinsic B-trees).*
16. *Better Deadlock detection and prevention.*
17. *Dynamic Transactions covering more than one IMAGE database (DMDBX).*
18. *IMAGE dataset capacity changes without having to DETACH / ATTACH.*
19. *Security enhancements in the IMAGESQL Utility program.*
20. *Enhanced UPDATE TYPE command in the IMAGESQL Utility.*
21. *DBSCHEMA and DBUTIL programs migrated to Native Mode (NM).*
22. *Make CIUPDATE = ALLOWED the default.*
23. *Provide a bundled 32-bit ODBC driver with Image/SQL.*
24. *Increase XM rollback limit from 2 MBytes to 4 MBytes.*
25. *DBINFO mode (406) to give fully qualified database name.*
26. *QUERY support of IMAGE intrinsic B-trees.*
27. *QUERY FIND, MULTIFIND, SUBSET: NO MATCH option.*
28. *IMAGE, DBUTIL, DBSCHEMA, DBChange Plus, and QUERY support dataset max entry length of 2378 words.*
29. *Multiple DBPUT - DBDELETE - DBUPDATE Dependency Semaphores: Independent Master-Detail scalability (DSEM).*
30. *QUERY FORM INDEXES to show all OMNI-DEX Indexes.*
31. *Make NM QUERY the default. Scheduled for MPE/iX 6.0 (soon).*
32. *DBUTIL option to confirm PURGE and ERASE (soon).*
33. *XM rollback limit > 4 MBytes mitigation (soon).*
34. *Increase internal limits to allow datasets > 40 GBytes (soon).*

## ***Are you a member of SIGIMAGE?***

Membership in SIGIMAGE (The INTEREX Special Interest Group for IMAGE/SQL Databases) is free.

You receive this SIGIMAGE Newsletter as a courtesy of INTEREX, the International Association of Hewlett-Packard Computer Users.

To ensure that you and your colleagues enjoy all the benefits of membership, contact INTEREX for further information:

INTEREX  
1192 Borregas Avenue  
Sunnyvale, CA 94088-3439  
U.S.A.

Telephone +1 (408) 747-0227  
Fax +1 (408) 747-0947

[membership@interex.org](mailto:membership@interex.org)  
<http://www.interex.org>