SIGIMAGE Newsletter

From the Chair

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

The year in perspective

1996 has been and continues to be another busy and productive year for SIGIMAGE. The kick-off event was a meeting of most members of the SIGIMAGE Executive Committee (SIEC) at the HP R&D Labs in Cupertino, CA on 2 February, to review CSY's direction for the HP 3000 and IMAGE/SQL. The positive outcome of this meeting set the stage for substantial progress on many fronts during the rest of 1996.

The next '96 milestone was the SIGIMAGE meeting at IPROF, which took place on 21 and 22 March in San Jose, CA. A total of 107 Image users and HP engineers signed up to attend two sessions that lasted $3\frac{1}{2}$ hours each.

This allowed users and HP to engage in extended discussions of the technical details involved in implementing the major enhancements now being worked on by the HP R&D Lab. IMAGE intrinsic B-trees (a.k.a. sorted sequential access), Master dataset dynamic expansion (MDX), and whether the CIUPDATE default should be ON or ALLOWED were especially popular subjects. Please refer to other articles in this newsletter by members of the HP R&D Lab team for much more information on these and other enhancements.

SIGIMAGE met again for two hours at HP World - Anaheim in August. At the strategic level, senior HP corporate managers said more nice things about the HP 3000 and IMAGE/SQL in public than has been heard by Interex conference attendees in a long time.

At the detail level, by making use of every minute and carrying on some discussions at the HP Database roundtable, IMAGE users managed to indulge in a little more technical tinkering with in-process IMAGE enhancement specifications, and added three new enhancement requests to the 1997 ballot.

As noted in the meeting minutes by Gary Biggs, a key administrative change for the SIG is that by majority vote of the attendees, from now on, election of officers will take place at IPROF instead of at the big conference in the fall.

By consensus of the SIEC and as confirmed during the SIGIMAGE meeting at IPROF, I served as interim SIGIMAGE Chair between IPROF in March and HP World in August. During the meeting at HP World I was formally elected as SIGIMAGE Chairman. Trying to fill the shoes that were so ably worn first by Steve Cooper and

then by Jerry Fochtman over the last six years is a pretty daunting task; I will clearly need the assistance of many of the other dedicated members of the SIEC and the entire SIG to try to pull it off.

Since you have already seen mention of something called the SIEC several times, it might be of interest to some readers to know what it does and who belongs. Turns out there is no mystery here: The SIEC is a group of some of the most technically knowledgeable long-time users of IMAGE, that have agreed to serve as the on-going liaison between SIGIMAGE and the HP R&D Lab.

This has proved to be very useful over the last several years, especially given the number and complexity of issues that have and will continue to come up while IMAGE/SQL is such a dynamic product undergoing major enhancement release cycles.

A total of 19 people including the Chair are currently on the SIEC distribution list, with six of these 19 members being HP employees. The size of the SIEC, the depth of IMAGE technical knowledge, and the years of experience represented therein are unique among Interex SIGs. Members of the SIEC have on previous occasions noted that we have probably reached our optimum operational state when it is hard to tell who is an HP employee and who is not. In that spirit the names only of the members of the SIEC are (although of course the rest of this newsletter is a give-away): Wirt Atmar, Jon Bale, Denys Beauchemin, Gary Biggs, Tien-you Chen, Steve Cooper, Bharati Desai, Jerry Focht-man, K N Kumar, Jeanette Nutsford, Shobha Pradeep, Kriss Rant, Alfredo Rego, Gilles Schipper, Stan Sieler, Ken Sletten, Rich Trapp, Fred White, René Woc.

Where we are now

The HP R&D Lab is on the verge of releasing of a number of major enhancements to IMAGE/SQL. These will follow closely behind other increased functionality that just achieved general distribution since August '96, with the 5.5 release of MPE/iX.

The 5.5 release includes support for:
 • TurboSTORE/iX 7x24 True Online Backup.
 • Indexed access through IMAGE/SQL.
 • Security enhancements in the IMAGESQL utility.
 • An enhanced UPDATE TYPE command.
 • Jumbo datasets.
 • Dynamic rollback of transactions spanning multiple Image databases (DMDBX).
 • Enhanced DBPUT and DBUPDATE integrity checking.

 • Migrating the DBSCHEMA and DBUTIL programs to native PA-RISC mode.

In addition, an ANSI SQL2 CAST function that will be included in the first 5.5 PowerPatch is available immediately as a patch to MPE/iX 5.5

A large amount of detail on all these enhancements is given on pages 6-39 through 6-72 of the MPE/iX Release 5.5 Communicator (HP part number 30216-90124).

Within the next few months users can expect to see the availability of the following additional major enhancements: A bundled 32-bit ODBC driver, intrinsic B-trees, MDX, and high-end scalability improvements.

Because of the need to complete, support, and manage the distribution of all these new features, the HP R&D Database Lab has not yet indicated what items they will focus on for follow-on enhancement releases. However, HP has indicated that the R&D Lab is funded and staffed, and that part of that funding is targeted for additional enhancements.

It is also a fact that since a resurrected SIGIMAGE produced the first big burst of enhancement requests in Reno in 1991, HP has for the most part gone ahead and implemented IMAGE enhancements pretty much in the order designated by the yearly SIG ballot. I am mentioning this in an attempt to impress all the readers of this newsletter with the importance of my next subject: What all SIGIMAGE members need to do next.

The 1997 SIGIMAGE enhancement ballot

Along with this edition of the SIGIMAGE Newsletter, you should have received the latest revision of the list describing all outstanding IMAGE/SQL Enhancement Proposals generated by the SIG, and your official 1997 Enhancement Ballot.

The list of enhancement proposals and the ballot have undergone quite a bit of revision for 1997, to remove the enhancements that are pending completion by the Lab and to add the new requests that were generated during 1996. In addition, the grouping has been changed and additional clarification has been provided in some of the descriptions.

If your site wants to be one of those who participate in the decision on what IMAGE/SQL enhancements HP will implement next, you need to take the time to review, fill out, and above all *return* your ballot so that it can be counted. The ballot can be returned by mail, by fax, or even by e-mail (if you get your copy of the Newsletter and the ballot by electronic means).

Please refer to the ballot for detailed instructions on when and how to send it back.

Note that each individual is only allowed to send in *one* enhancement ballot, regardless of the return method. To try to help keep the one-person, one-ballot rule straight, for an enhancement ballot to be counted, it must have at least the name and the phone number or -mail address of the person sending it in; and that information must be legible.

Remember that you have a total of *only* 100 votes to spend on the ballot. And keep in mind that these are *absolute* votes; i.e., you can vote against an enhancement by prefacing the vote amount with a minus sign, but if you do that, you do *not* get 100 positive votes plus some amount of negative votes (someone tried to do that last year). The numbers have to add up to 100 regardless of the sign!

Coming attractions

SIGIMAGE will of course meet again at IPROF-97. The dates and location for the next IPROF have not been locked in yet, but are expected to be available soon. Watch for announcements on the Internet on HP3000-L, on various HP 3000 vendors' web sites, in Interex-PRESS, in the 3000 News/Wire, and in direct hardcopy mailings from Interex.

IPROF should be especially interesting in 1997, since it will be the first general meeting of HP 3000 users following the expected announcement by HP in January of their five-year plan for the HP 3000. Given the generally positive verbal trend at HP World in August, we are looking for more good news and action by HP in the near future.

Acknowledgments

Special thanks to my predecessor Jerry Fochtman, for his years of leadership and many hours of work on behalf of SIGIMAGE and all IMAGE users. SIGIMAGE will continue to benefit from Jerry's extensive knowledge of IMAGE/SQL, since he has agreed to continue to serve as a member of the SIEC.

Kudos to the HP R&D Database Lab, who took the time to produce all the articles they contributed to this issue of the Newsletter. I believe this is a new HP record for one edition. Thanks also to Alfredo, who again stepped forward and volunteered to edit and publish another issue.

All my fellow members of the SIEC, for their contributions of time and effort. Without them SIGIMAGE would be much less successful than it is.

Wirt Atmar, the rest of the team at AICS Research, Adager, and everyone else who participated in the HP 3000 World's Largest Poster Project at HP World in Anaheim.\

And of course thank you to all the thousands of users around the world who continue to bet their businesses on IMAGE and the HP 3000. Just remember to take the time to fill out and send in those 1997 enhancement ballots :-)

Report from the Lab

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

It was nice to see many of you at HP World '96 a few weeks ago. We not only encountered each other at the conference and show; I also caught a number of familiar faces in the "home video" footage I shot at the construction of the world's largest poster!

For those who did not attend the conference, here's a brief update on the status of our enhancement projects in the lab. Further information on these projects and other topics is given in accompanying articles written by members of our database lab team.

Of course, we are devoting a substantial amount of effort to product maintenance and hot sites (believe it or not, we have some!) and (until recently) to final integration and test of IMAGE/SQL and ALLBASE/SQL with MPE/iX 5.5, and ALLBASE/SQL ported to HP-UX 10.10. Having a lab team in Bangalore has really helped us to complete a number of these tasks.

One of the key components in the MPE/iX 5.5 release is TurboSTORE/iX 7x24 True-Online Backup. As you probably know, we made some enhancements in IMAGE/SQL to facilitate this new backup capability, including changes in DBRECOV to allow successful IMAGE database recovery based on TurboSTORE backup tapes. Kumar's article provides more details.

Another recently completed feature, this one in SQL (benefiting users of IMAGE/SQL and ALLBASE/SQL), is the CAST function. This function allows explicit conversions of data from one data type to another, for a variety of supported data types. Doug Myers gives more information about CAST in his article.

As we announced at HP World, we have changed our implementation strategy for the new 32-bit ODBC driver. Instead of completing our own lab project, we have decided to work with M. B. Foster Associates to bundle and ship with ALLBASE/SQL and IMAGE/SQL a special version of the ODBCLink option of their DataExpress product.

This approach has two major benefits for HP and our customers.

• First, consistent with CSY's strategy of working closely with third parties, it enables us to focus our in-house R&D resources on those high priority projects which only HP can do.

• Second, our ODBC component is being designed to allow an easy upgrade path for customers who wish to purchase the increased capabilities of the full DataExpress/ODBCLink product from M. B. Foster Associates.

Additional information on our new 16-bit and 32-bit ODBC drivers is contained in an article by Shu-Feng Wei.

Our B-tree implementation is progressing well, including needed changes in the TurboIMAGE and IMAGE/SQL components.

We made some modest design changes after presenting our plans at IPROF, based on the feedback we received at that event and in the weeks immediately after. This really reinforces for me the value of working closely with the SIGs on our designs. You will find an article by Bharati Desai in this newsletter providing more details on our design.

We were able to demonstrate our B-tree functionality in action at HP World, and as I write this in early September we are just about finished coding the last modules of the full enhancement. We will be beginning full-fledged testing later this month, and all testing should be complete before the end of the year. If you are interested in participating in our beta testing of B-trees, please contact Bharati Desai (bharati@cup.hp.com).

The dynamic master dataset expansion (MDX) capability, an enhancement originally committed last year as part of our High Availability Solution, will be restarted within a few weeks. The software development on this one is about half done, and my hope is that we can complete it by the end of the year. As of now, I can't give you a definitive testing and release schedule, but we intend to get this capability to you as soon as we can in 1997.

As I mentioned at HP World, we are also working on some internal performance enhancements for the Platform Evolution Solution that we refer to as *IMAGE Scalability.* The idea is to remove or reduce the bottleneck effect of the IMAGE database "put-delete semaphore" that seems to be emerging with the introduction of higher performance systems. These changes, if our tests show them to yield increased throughput, will be included in a future release aimed at supporting new high-end systems (as well as all existing ones).

There's also an article by Tien-You Chen in this issue called *DBUTIL Tips.* Tien-You explains our plans to change the default setting to ALLOWED for the flag known as CIUpdate (critical item update), as well as describes how to use the DETACH command in DBUTIL to overcome a problem that can arise when you are trying to restructure an IMAGE database.

Finally, let me give you a travel report. In the Winter 1996 issue of the SIGIMAGE Newsletter, I mentioned that Kriss Rant of CSY Marketing and I were about to leave on a trip to our Bangalore, India lab.

I can tell you now that we had a tremendously productive and enjoyable visit. Our host in Bangalore was V S Subrahmanyam—we call him Subbu—, our CSY lab manager there. You may have met Subbu during one of his U.S. visits or other travels.

CSY has a very good (and improving) communications infrastructure linking the Cupertino and Bangalore teams, but nothing can replace face-to-face meetings for getting to know people and establishing a basis for working together.

My DB team in Bangalore is great, and I am very pleased to know each of the members personally now. They are highly-qualified engineers working in well-equipped (though crowded due to recent growth) offices and labs. The city of Bangalore can be described in many ways—green, growing, bustling, exotic, noisy, and friendly come to my mind. Many high-tech companies are located there; one building in which HP is renting space also contains offices of Microsoft, Silicon Graphics, and a large number of other companies.

After returning to California, I found myself on the HP internal "lecture circuit," showing my travel video to many of our people here who work closely with the folks in India. These presentations were helpful, I think, in linking the teams in a new way.

My show is getting old, however. In the months since my trip, many more engineers and managers have traveled between the two sites, so there are many more close linkages now. Moreover, CSY Bangalore is just about to move from two neighboring buildings into one brand new building, so my video will be sadly out-of-date. I hope to do an update sometime in the next few months!

That's my trip report in a nutshell. Kriss Rant's visit was equally fruitful, I believe. You'll have to get the details directly from him; be sure to ask him about the monkeys!

1996 SIGIMAGE Enhancement Ballot Final Results

Ken Sletten
SIGIMAGE Chairman
NUWC Division Keyport
Keyport, WA, U.S.A.

Following is the final "Top 20" ranking from the 1996 SIGIMAGE Enhancement Ballot, including ballots received from the European International Users Group (IUG) and those sent back from the Interex Application Developers Survey mailing.

A total of 65 valid ballots were received. Each voter was allowed to "spend" 100 votes on his or her ballot, divided up in any desired proportion between one or more enhancements. Note that the IUG and Developers Survey ballots resulted in a slight change in the ranking from what was published on the HP3000-L Internet discussion group after IPROF-96 and at HP World.

The clear winner from the '96 ballot was to have HP provide a bundled 32-bit ODBC driver for Image/SQL. Responding to this request, HP is currently expected to deliver this enhancement by the end of 1996. Please refer to other articles in this newsletter by Jon Bale and Shu-Feng Wei for additional implementation details on 32-bit ODBC.

Also note that item 32 on the 1996 ballot (Make CIUPDATE = ON the default) was discussed at length and with some passion at IPROF-96. Final consensus was that making CIUPDATE = ON the default might result in backward incompatibility for a very small number of users. To avoid any possible problems, it was collectively decided that making CIUPDATE = ALLOWED the default was the safest thing to do. Please see another article in this newsletter by Tien-You Chen for additional discussion of this issue.

In the interest of conserving space in this extended edition of the Newsletter, the 1996 ballot results are presented in compact format. In addition, only the Top 20 ranking is listed, out of a total of 44 items that were voted on (38 were on the printed ballot that was mailed out; six more items were added and voted on at IPROF-96).

A short description of each enhancement is preceded by three "tagged" numbers: "#dd, (ddd), [dd]". The digits after the "#" designate the 1996 ballot ranking; digits between the "()" give the number of votes for each; digits between the "[]" denote the item number as it appeared on the 1996 ballot.

01, (685), [35]. Provide a bundled 32-bit ODBC Driver for IMAGE/SQL.

02, (556), [20]. Extend IMAGE/SQL data access to include MPE and KSAM files.

03, (486), [01]. Ability to add/drop datasets, indexes, and items on the fly.

04, (385), [22]. Provide Date/time datatype in IMAGE/SQL.

05, (378), [38]. Provide ability to FIND FIELDA = FIELDB in QUERY.

06, (299), [21]. Provide Image/SQL support for Binary Large Objects (BLOBS).

07, (281), [09]. Provide ability to retain and reuse attach parameters across a detach/re-attach of an IMAGE/SQL database to a DBE.

08, (254), [05]. Support SQL NULL items in IMAGE/SQL.

09, (249), [12]. Provide a user-accessible DBQUIESCE intrinsic for IMAGE.

10, (242), [15]. Provide a Measurement Interface for IMAGE performance data.

11, (224), [14]. Allow tracking files to be associated with IMAGE/SQL databases.

12, (215), [19]. GETUPDATE to update record(s) without prior DBGET.

13, (199), [02]. Provide a read-only DBLOCK mode (repeatable read) in IMAGE.

14, (194), [23]. Provide the ability to specify flags and settings in DBSCHEMA.

15, (170), [36]. Provide a MULTIFIND update/delete function in QUERY.

16, (141), [08]. Improve DBA functions for managing authorization IDs.

17, (133), [10]. Provide ability to enable/disable SQL access without having to ATTACH/DETACH an IMAGE database to/from a DBE.

18, (123), [16]. Provide DBCONTROL access in QUERY.

19, (100), [03]. Allow multi-record/multi-set DBGET, DBPUT, & DBUDPATE.

20, (083), [06]. Support SQL SAVEPOINT in IMAGE/SQL.

DBUTIL tips

Tien-You Chen
Hewlett-Packard Company
Cupertino, California, USA
Bangalore, Karnataka, India

CIUPDATE setting

In the SIGIMAGE meeting held at IPROF 1996, the attendees decided that the default for the CIUpdate flag in the rootfile should be "ALLOWED" instead of "DISALLOWED," so more users can take advantage of the enhancement.

- Setting the CIUpdate flag to ON means all programs accessing the database can update detail critical item(s) with the DBUPDATE intrinsic.
- Setting the flag to ALLOWED means programs will not be able to update detail critical item(s) with the DBUPDATE intrinsic *unless* a one-time DBCONTROL mode 5 is done prior to DBUPDATE. This programmatic setting remains into effect for the duration of the process until the end of the program or until it is subsequently disabled via DBCONTROL mode 6. This retains backward compatibility, so old programs will work as usual.

For IMAGE users, changing the default setting of DISALLOWED to ALLOWED is a one-time step using a version of TurboIMAGE/iX that has this feature. Let us assume that this feature will be available in Version C.07.00.

During the first DBOPEN of a database using TurboIMAGE/iX version C.07.00:

- If the setting is ON or ALLOWED, it will remain unchanged.
- If the setting is DISALLOWED, DBOPEN will change it to ALLOWED.

This is fine for users who want the ON or ALLOWED setting. What if you really desire DISALLOWED? You need to set it explicitly to DISALLOWED, once, using the SET command of DBUTIL version C.07.00. This can be done before or after the first DBOPEN automatically changes it to DISALLOWED. For example:

```
SET basename CIUPDATE = DISALLOWED
```

Once you set the CIUpdate flag to DISALLOWED explicitly (using version C.07.00), it will remain into effect until you decide to change it again.

DETACH command

You may experience the following situation: You have a production database attached to an ALLBASE DBE and you want to restructure your database. To avoid disrupting your normal operation, you make a copy of the database (including the TC file) and restore it to another group/account, or even to another machine. Unfortunately, when you start to restructure this copy of your database, the program informs you that the database is attached. You get an error because you can't restructure the database while it is attached. Subsequently, you may try to run IMAGESQL.PUB.SYS to detach, but IMAGESQL tells you that the copy of the database is not attached to the original ALLBASE DBE. You are stuck!

Fortunately, DBUTIL now has a DETACH command which can bail you out from this situation. You just need to type the following to solve your problem:

```
>>DETACH basename
```

You may get an error stating that the detach failed from `dbename.group.account` with error number `(ATCERR 32052)`. This happens because the copy of the database triggers a lookup of the TC file (which contains the names of the DBEs to which the database is attached). However, DBUTIL's subsequent verification of the *original* DBE shows the name of the *original* database (not the name of the *copy* of the database) as the one attached to this DBE. Due to this discrepancy, the following things happen:

- DBUTIL reports an error and does not perform a real detach (of the copy);
- in spite of the error, the ATTACHed flag in the rootfile (copy) as well as the entry in the TC file is cleared.
- the original production database still remains attached to the DBE.

So now you can perform your restructuring procedures on the copy of your database!

Well, the above scenario is for a copy of a database. In normal circumstances (when you work with a database that was originally attached to a given DBE or to several DBEs), the DETACH command works correctly to detach the database from the specified DBE(s). Actually, the DETACH command in DBUTIL becomes a handy shortcut if you have several DBEs to which the database is attached. One command of DBUTIL detaches a given database from all of the DBEs to which it is attached; you do not have to specify each DBE name, one at a time, while you are in IMAGESQL.

Can you FTP a database?

F. Alfredo Rego
Adager Corporation
Sun Valley, Idaho 83353-3000 U.S.A.

Patrick Anderson asked this question on HP3000-L, the Internet discussion group. James B. Byrne commented:

Of course, as this is an MPE/iX product (FTP3000), there is absolutely nothing stopping HP from extending the command set to include a DBMGET which operates similarly to a MGET but only takes as a valid argument the root file of an IMAGE database and contains the privileged-file handling code.

After careful consideration, Ken Sletten (SIGIMAGE Chairman) included Jim's proposal in the upcoming SIGIMAGE enhancement ballot that will accompany this issue of the SIGIMAGE Newsletter.

B-trees in IMAGE/SQL

Bharati V. Desai
Hewlett-Packard Company
Cupertino, California, USA
Bangalore, Karnataka, India

The need for generic key searches and range searches has been recognized by IMAGE customers for quite sometime. This is why the request for B-trees in IMAGE/SQL has consistently been the most desired enhancement of the SIGIMAGE list.

Well, Jon Bale briefly mentioned the status of this enhancement and its availability in his project update article in this newsletter. I now discuss the design, externals, and some high-level internals.

Continuing with the prevailing tradition in the IMAGE Lab, the input for its external design was obtained from several customers during (and after) IPROF '96. We believe that the final implementation addresses the primary needs of the majority of our IMAGE customers. Our objective has been to provide this new feature and also give you as much flexibility as possible to control the use of this feature.

The highlights of this enhancement are:
- You can create a B-tree *only for the key item of a master dataset.* Nevertheless, you are able to perform a B-tree search for all of its corresponding detail sets as well. A master dataset is perceived as having an explicit B-tree and a detail dataset is perceived as having an implicit B-tree for each of its paths.
- You may create B-tree indices for zero, one, or more master datasets.
- DBSCHEMA introduces a new option, INDEXED, for the "set" specification.
- DBUTIL introduces new commands and options.
  - New ADDINDEX, DROPINDEX, and REBUILD-INDEX commands (to be used for one, more than one, or all masters).
  - The SET command allows you to define your own wildcard character to indicate "match any trailing text". The default wildcard character is "@". The concept of "#" and "?" (for single numeric and alphanumeric character, respectively) is not available. You can specify a generic search using this wildcard character as part of your DBFIND argument when the search item has a B-tree (explicit or implicit).
  - New option, BTREEMODE1, for the SET command to set DBFIND mode 1 access ON or OFF for a B-tree search for X and U types. The default is OFF, in which case DBFIND mode 1 will continue to work the way it does currently, even when the argument contains a wildcard character! When BTREEMODE1 is ON, DBFIND mode 1 (with a wildcard character in the argument) will be treated as a B-tree search.
  - The CREATE, ERASE, PURGE, SECURE, RELEASE, MOVE, and SHOW commands include B-tree files in their operation.
- DBCONTROL
  - Mode 13, to be used with the qualifier, to allow privileged-mode (PM) applications to create and maintain B-tree indices.
  - Mode 15 is used to set the BTREEMODE1 option to "on" on a database-wide basis and to specify the wildcard character.
  - Mode 16 sets the BTREEMODE1 option "off" on a database-wide basis.
- DBINFO
  - Mode 209 informs whether or not a B-tree index exists for a master.
  - Mode 113 gives you the BTREEMODE1 setting and the wildcard character.
- DBPUT/DBDELETE to an indexed master triggers a similar operation to the indexed master's B-tree file.
- DBFIND
  - DBFIND can be used for details as well as for masters to specify B-tree searches.

- For wildcard (generic) searches, the wildcard has to be the terminating character in the argument. Character(s) beyond that will be ignored.
- If BTREEMODE1 is ON, DBFIND mode 1 will result in a B-tree search if:
  - the dataset has a B-tree index;
  - the item is of type X or U;
  - there is a wildcard character in the argument.
- DBFIND introduces new modes 4, 10, 21, and 24.
  - Modes 4 and 24 offer a new argument construct to include ">", ">=", etc. for range searches.
  - Modes 21 and 24 are faster versions (they do not return accurate chain counts) of modes 1 and 4 respectively.
  - Mode 10 allows you to simulate the current DBFIND mode 1 even when the dataset has a B-tree, BTREEMODE1 is ON, and the argument contains a wildcard character.
  - The status array reflects information based on dataset, mode, and search type. B-tree searches with modes 1 and 4 for details give superchain (multiple detail chain) counts and pointers to the first entry in the first detail chain and to the last entry in the last detail chain. For masters, only the count of qualified entries is returned for modes 1 and 4 (pointer information is set to zeroes).
- DBGET modes 5 and 6 can be used for master B-tree searches as well as for details. Superchains are traversed for details.
- There is one KSAM/iX privileged file for each B-tree. The file-system size limit for this B-tree file is 4GB. A Jumbo master (larger than 4GB) can have a B-tree, provided the B-tree remains within its 4GB limit.
- The KSAM/iX B-tree file is named using the POSIX file format with the "idx" extension. Example: /ACTSALES/GRPSALES/ORDERS03.idx
- Only the full master key value (without pointer information) is written to the KSAM/iX file.
- Record zero of the root file contains "C4" for the root file version if at least one B-tree exists for the database. When all B-trees are dropped, it reverts to the appropriate version—"C3" if at least one Jumbo set exists, "C2" otherwise.
- A new bit map is added in the root file for B-trees.
- During ATTACH, the SQL Optimizer is made aware of B-tree indices which can be found in the SQL table, SYSTEM.INDEX, of the SQL catalog. B-tree indices are registered *only* in the DBE to which the database is ATTACHed. For a master, a B-tree is registered as a UNIQUE index; all of its corresponding detail search items are registered as having NON-UNIQUE indices.

- Third-party indices can coexist with B-trees.

Remember that, with more indices, there is more overhead for puts/deletes/updates, as well as increased complexity for the SQL Optimizer (because the INDEX for the query is chosen by the SQL Optimizer and cannot be overridden by the user.)

From the above highlights, it is easy to see that the B-trees enhancement can accomodate varied needs of a customer in the same computing environment. Furthermore, there are many ways you can start using this enhancement.

For sophisticated use of B-trees, especially for range searches for numeric items, you will need to make application changes. More detailed information on advance use is beyond the scope of this article. However, if you desire *generic search* only and want to get started quickly without making any application changes, you may employ the following approach.

How to get started quickly

First of all, you must identify the masters which need B-trees. Then, you must create B-trees using one of the following two methods:

For new databases

Use the new INDEXED option of DBSCHEMA.

Syntax:

```
Name: setname, <manual | automatic>
    [/indexed]
    [(read-class-list)/(write-class-list)]
    [,device-class];
```

Example:

```
Name: Employeename, manual /indexed
    (10,20/30);
```

For existing databases

Use the new ADDINDEX command of DBUTIL.

Syntax:

```
addindex basename [/maintword]
      for <all | setnamelist | setnumlist>
```

where `setnamelist := setname[,...]` and `setnumlist := setnum[,...]`

Examples:

```
>Addindex orders/secret for ALL
>addindex orders/secret for 1,7
>Addindex orders/secret for SalesrepName,
 Region, District
```

BTREEMODE1 option

Set the BTREEMODE1 option ON using DBUTIL as follows:

Syntax:

```
set dbname[/maintword]
    BTREEMODE1=< on | off> [,[wildcard=]c]
```

where "c" is any printable ASCII character. Notice that "wildcard=" is optional (i.e., you can type the wildcard character by itself).

Examples:

```
set orders/secret BTREEMODE1 = on
set tape/pass BTREEMODE1 = on, @
set cust/boss BTREEMODE1 = on, wildcard = *
```

That's all (for a quick start)

Now you are ready to perform a B-tree search. If your database is new, you will need to add data. After that, you can include the wildcard character in your DBFIND argument and start observing the results.

Are you hungry for more?

The above method is the quickest and simplest way to benefit from B-trees. More information on advanced use of B-trees will be made available later via different media.

Are you interested in beta testing?

If you are interested in beta testing this feature, please send me an e-mail message, copying your HP support person:

```
bharati@cup.hp.com
```

Understanding "Prefetch" in IMAGE

Ken Paul
Adager Corporation
Sun Valley, Idaho 83353-3000 U.S.A.

Randy Smith posted the following on the HP3000-L Internet list/newsgroup:

Would someone explain TurboImage Prefetch? I was reading Robelle's "What's Up DOCumentation", which talked about increasing TurboIMAGE performance by using prefetch.

- I can infer many things from the term prefetch as applied to a database, but what does it mean exactly for IMAGE?
- What are the benefits?
- What are the drawbacks?
- Are there any gotchas?

Background.

Prefetch is a flag that can be *enabled* for your database via DBUTIL. It was introduced in TurboIMAGE version C.03.13, which corresponds to MPE/iX version 3.1. By default, this flag is *disabled* for your TurboIMAGE databases and must be explicitly *enabled* using DBUTIL.

The term "prefetch" can have many different meanings. Usually, prefetch refers to file I/O, where the system is going out to disc to get more information before you ask for it. This type of prefetch can greatly enhance the performance of your programs, especially if the data that is being prefetched by the system is indeed the data your application needs. This usually is very beneficial with a *single* process that is doing serial reads of a file or dataset (this is typical in batch mode, for instance).

Unfortunately, this is not what is meant by the TurboIMAGE prefetch flag.

TurboIMAGE is currently *single threaded* when it comes to doing DBPUTs, DBDELETEs and, most recently, Critical Item DBUPDATEs. This means that only *one* process is actually adding or deleting (or updating a critical item) at a time. In order to accomplish this, TurboIMAGE uses a semaphore lock known as the PUT/DELETE semaphore. This lock is *applied* before the updating of several blocks within the database and *released* at the end of the updating, to ensure data integrity.

The prefetch flag tells TurboIMAGE when to apply this semaphore lock.

- Before the existence of the prefetch flag (and with prefetch disabled), TurboIMAGE locks the PUT/DELETE semaphore before reading, updating, and writing every data block which is modified by the transaction. After this is done the PUT/DELETE semaphore is unlocked.

- With prefetch enabled, TurboIMAGE reads all the data blocks that are needed for the complete transaction before the PUT/DELETE semaphore is locked. Once the semaphore is locked, TurboIMAGE updates and writes the data blocks which are modified by the transaction before unlocking the semaphore.

It is the holding of the semaphore lock for a shorter period of time which could improve the throughput of data on your system, but this is not guaranteed. There is the possibility that after the data is prefetched—and before the semaphore lock is applied—another process may flush it out of memory before TurboIMAGE is able to update it. If this is the case, TurboIMAGE must go out to disc to re-read the block and this extra I/O activity will have a negative impact on your database performance.

Effectiveness of prefetching

The Prefetch flag has always been a "your mileage may vary" flag since its introduction. Certain conditions need to exist on your system in order to—possibly—gain benefit from turning on this flag.

First, the system should have "adequate" (a wonderfully nebulous word) memory available to handle the increased data page locality, as well as adequate processor capability to handle increased concurrency of processes.

Second, applications should make numerous calls to DBPUT, DBDELETE (or, most recently, DBUPDATE to critical items).

Third, multiple processes must be actively updating a database before a benefit can be realized.

The HP Database Lab has said that they have seen 5% to 10% improvements in the number of transactions that could be processed in benchmarks as a result of enabling prefetch. They also mention that it helps in situations where you have excess CPU time and you are running update-intensive applications (I think they mean intensive in terms of DBPUT/DBDELETE/Critical-item DBUPDATE).

The 35% performance improvement that is mentioned in Robelle's "What's Up DOCumentation" is news to me. I have never seen or heard of such a high value. I consider the prefetch flag a "try it, you might like it" flag.

Problems with prefetch

There was a problem with prefetch under TurboIMAGE versions C.04.06 through C.04.08, when DBPUTs to a detail dataset were failing because this flag was set and there were some uninitialized variables which caused problems. This bug, however, did *not* cause database corruption. The fix was to disable the database for prefetch until the next version of TurboIMAGE was released via a patch.

I also ran into a site which was experiencing bizarre "DBG disabled" messages while opening a database. The messages could not be traced back to broken chains. After analyzing the database and finding no broken chains, we noticed that prefetch was enabled and, upon disabling it, the problem disappeared. This was on TurboIMAGE version C.04.19.

My advice

Enable the prefetch flag and see if your environment improves noticeably. You can always turn prefetch off quite easily if any problems appear.

Gratitude

I would like to thank Marguerite Bryan of the HP Response Center and Jim Sartain, formerly of the HP Database Lab, for information which I used in this article.

DBRECOV and True-Online Backup

Kumar KN
Hewlett-Packard Company
Cupertino, California, USA
Bangalore, Karnataka, India

DBRECOV, the TurboIMAGE database recovery utility, has been enhanced to perform recovery of databases stored using the new TurboSTORE/iX 7x24 True-Online Backup. This feature in DBRECOV (version C.06.10) is available starting with MPE/iX 5.5 (and TurboIMAGE/iX version C.06.12).

MPE/iX 5.5 has several new features to meet your High Availability needs. One of them is True-Online Backup, which not only quiesces the database if open for read/write access, but also stores your entire logical database including the TC file, and third-party index (TPI) files, if any. (True-Online Backup also stores Jumbo datasets.) With this enhancement to DBRECOV, you can make full use of True-Online Backup for databases and have the same reliability and recovery capabilities as before.

True-Online Backup provides full-featured online backup capabilities for a 7x24 environment. It allows users to continue running their applications while the backup occurs. For databases, this means that they can be open for access during the backup. In order to perform a True-Online backup, TurboSTORE/iX requires that one of its two available options for sync point (ONLINE=START or ONLINE=END) be used in the store command. The sync point of a backup refers to

the point in time during backup when the database is logically consistent. The sync point can be at the beginning or at the end of backup. Please refer to the 5.5 Communicator for details.

TurboSTORE will automatically quiesce the database(s) and synchronize all updates to the database(s) to maintain logical database integrity.

For TurboIMAGE databases, quiescing implies that all open transactions are allowed to complete before the sync point for backup is enforced. New transactions are allowed to start after the sync point. For ALLBASE databases, all open transactions are rolled back to enforce the sync point. In either case, it is ensured that the database is in a logically consistent state for backup purposes.

Logging, if enabled for a database, need not be suspended before the database is backed up. This also means that no new log cycle needs to be started after the backup. Therefore, the recovery, when applied, will commence from the middle of the log file. Also, the log file need not be the first one in the set. The name of the starting log file required for recovery is stored in the root file and can be viewed with the "SHOW database ALL" command of DBUTIL. This will enable you to determine which log files in the log file set are not needed for recovery and, hence, may be purged. If the database is enabled for logging, it is necessary for the log process to be active at the time of True-Online Backup. This will be required whether the database is open or not at the time.

No application modification is necessary to take advantage of this new feature of DBRECOV.

DBRECOV will identify the correct log file to use for recovery and will also determine the transaction from which to start the recovery and proceed from there. If True-Online Backup has not been used to store the database, DBRECOV will start with the first log file in the log file set. Furthermore, DBRECOV will continue to work with databases stored using other utilities such as DBSTORE.

Copyright information

Explicit Data Type Conversion (CAST)

Doug Myers
Hewlett-Packard Company
Cupertino, California, USA
Bangalore, Karnataka, India

With the G1.15 Version of ALLBASE/SQL (or the ALLBASE/SQL component of IMAGE/SQL), the CAST function enhancement is now available to both ALLBASE/SQL and IMAGE/SQL customers.

The CAST function is used to explicitly convert data from one data type to another. The CAST function not only allows conversion between compatible data types, such as between CHAR and BINARY or between INTEGER and DECIMAL, but it will also allow conversion between certain normally incompatible types, such as between CHAR and INTEGER.

The CAST function is defined in the ANSI SQL2 standard. CAST in ALLBASE/SQL and IMAGE/SQL complies with that standard. In addition, several extensions to the standard specification have been added to make CAST even more powerful.

The CAST function can be used anywhere a general expression is allowed. Also, as a part of this enhancement, the SQL parser has been enhanced to allow general expressions in more of the SQL syntax. For example, general expressions including nested functions are now allowed in all the date/time functions and string functions. Therefore, CAST will be supported inside functions that support expressions, including aggregate functions. CAST will also take general expressions, including nested functions, as input.

CAST Syntax

```
CAST ({<expr>}{AS}<datatype>[,<formatspec>])
     {NULL} {,}
```

`<expr>` can be a column, USER function, host variable, local variable, AddMonths function, aggregate function, date/time conversion function, dynamic parameter or procedure parameter, constant, current function, long column function, string function, or any combination of these in an arithmetic or concatenation expression

`<datatype>` is an ALLBASE/SQL data type: CHAR(n), VARCHAR(n), DECIMAL(p[,s]), FLOAT, REAL, INTEGER, SMALLINT, DATE, TIME, DATETIME, INTERVAL, BINARY(n), VARBINARY(n). The LONG BINARY(n) and LONG VARBINARY(n) can not be used in the CAST operations.

`<formatspec>` is for DATE, TIME, DATETIME, INTERVAL conversion. `<formatspec>` is the same as that used in the date/time conversion functions. See the ALLBASE/SQL Reference manual for more details.

For more information on CAST, including examples, please refer to `http://jazz.external.hp.com/papers/Communicator/dbGl/cast/cast.html` on the World Wide Web.

IMAGE date-oriented items and the next millennium

F. Alfredo Rego
Adager Corporation
Sun Valley, Idaho 83353-3000 U.S.A.

The year 2000 and beyond

Most computer systems will have a hard time after December 31, 1999. Fred White, René Woc, and I have done a lot of work regarding the examination and conversion of date-oriented information that resides in IMAGE databases.

In this article, I report the results of our research and development, using Adager as a specific platform. In a previous issue of the SIGIMAGE newsletter, Kriss Rant described the research and development efforts that are going on within Hewlett-Packard. It would be very interesting for our readers to see other reports on this timely topic in future issues of the SIGIMAGE newsletter. So, I hereby extend an open invitation to other individuals to share their approaches to the millennium challenge.

Examining dates

We allow you to produce an exhaustive report of potential date-oriented problems. You can verify the contents of your date fields, even if you don't have any immediate need to convert these values for the next millennium.

The *Examine Date* command can log all dates which fulfill certain "exception" criteria that you may specify (such as null, zero, blank, or invalid). If you have overloaded date values, you can ask Adager to accept them. You can also log all dates within (or outside of) a given range.

This functionality is of great value to "inventory" your date-oriented values, whether you decide to convert them or not.

Converting dates

Armed with the information provided by *Examine Date,* you can easily configure Adager's *Change Date* command, which converts the formats of your date-oriented data items:

- in the privileged internal tables that reside in the database's root file;
- in the data entries that reside in the datasets where the involved data items are defined as fields; if some of these data items are defined as *search fields* in master datasets, we automatically rehash the affected master datasets.

Source date-oriented formats

We handle ASCII as well as binary date-oriented values in a variety of formats. As soon as you specify the source data item, We report all its possible date-oriented formats.

For convenience, let's say that "c" refers to a century, "yy" refers to the two digits of a year, "mm" refers to the two digits of a month, "dd" refers to the two digits of a day within a month, and "ddd" refers to the three digits of a day within a year.

There are date formats that are character-oriented. We support a variety of source character formats, including:

> yymmdd & yy/mm/dd
> mmddyy & mm/dd/yy
> ddmmyy & dd/mm/yy
> yyddd

There are date formats that include one digit for the century. We evaluate cyymmdd as ccyymmdd, where $cc = c + 18$.

There are date formats that are binary-oriented. We support a variety of source binary character formats, including:

> yymmdd
> mmddyy
> ddmmyy
> yyddd
> dddddd the number of days since January 1, 0000 (or since January 1, 1900 for DataExpress, or since January 1, 1973 for MANMAN)

If you desire, we can treat the value yymm00 as yymm01 and the value yymm as yymm01.

There are "proprietary" bit-packed formats. We support PowerHouse, MM/3000, HP Calendar, SRN Chronos formats, and 128-bit IMAGE/SQL (ALLBASE/SQL) formats.

In the case of MM/3000, the leading yy "digit" may be "A" through "Z" to mean a year beyond 1999.

How about overloaded date values?

Some applications may depend on special date values (which are, obviously, illegal dates) whose presence triggers some specific action on the part of the software.

For instance, 000000 may mean *not yet paid,* 870245 may mean *there were no new students this month,* and so on. Creative people have used all kinds of overloading tricks, including the use of blanks, binary zeroes, asterisks, month values beyond 12, etc.

We decided to reject all such illegal dates by default. Nonetheless, by design, we let you specify which of these special date values you want to explicitly allow, if any (and which type of conversion you want to apply to them, if any, so that your application software will require the minimum amount of changes).

Defining the current (or the next) century via a threshold or cutoff value

For those date formats that allow it, we permit you to specify a threshold value "tt" that will define a destination century of 19 if the yy component of your date is greater than (or equal to) tt and will define a destination century of 20 if the yy component of your date is less than tt.

For instance, if you specify tt as 77, a source date equal to 810212 will define a destination date of 19810212, whereas a source date of 660515 will define a destination date of 20660515.

Destination date-oriented formats

If you decide to convert your date formats with *Change Date*, we present you with a comprehensive list of target formats.

In particular, we only allow target date formats that contain a 4-digit year (implicitly or explicitly). So, you can never *downgrade* your date formats: We only allows you to *upgrade* them.

Can you improve your date sorting?

Yes. You can use *Change Date* to convert your dates from a sort-unfriendly format to a sort-friendly format, in addition to enhancing the year information for the next millennium.

For instance, you can easily convert data values in ddmmyy or ddmmyyyy formats to the sort-friendly yyyymmdd format.

HP's New ODBC Driver

Shu-Feng Wei
Hewlett-Packard Company
Cupertino, California, USA
Bangalore, Karnataka, India

Hewlett-Packard's Commercial System Division is preparing to release a new implementation of the ALL-BASE/PC API product. This is a revised OEM version of ODBCLink from M.B. Foster Associates, Ltd.

The name of the HP driver (to be bundled with ALL-BASE/SQL and IMAGE/SQL) has not been decided.

This PC API product is an implementation of Microsoft's Open Database Connectivity (ODBC) protocol for HP3000 and HP9000 data sources. It runs on Intel platforms in the 32-bit environment (MS Windows 95 and NT) as well as in the 16-bit environment (MS Windows 3.1 and Windows for Workgroups 3.11). Compatible access will be provided to IMAGE/SQL databases on HP3000 computer systems and to ALL-BASE/SQL databases on HP3000 and HP9000 computer systems. The older "Gupta" interface will not be supported. Communications between client and server use TCP/IP protocol links via the Winsock API.

The ODBC driver for 32-bit and 16-bit Windows environments is being developed, tested, and certified with Microsoft's and WRQ's implementations of the `winsock.dll` protocol library.

The ODBC Driver is being developed to conform to the Microsoft ODBC 2.0 specification.

- API Conformance Level: Fully implements all Core, Level 1, and Level 2 API calls (except extended fetch and its related calls).
- SQL Conformance Level: All Core, ODBC 2.0 Extended Data Types, and ODBC SQL escape clauses.
- ALLBASE/SQL and IMAGE/SQL interoperability:
  - Multiple Active Connection-Statement-Transactions from a single application or from multiple applications.
  - Stored Procedures, Outer Joins, String Functions, Blobs, Data definitions.
  - Cursor characteristics:
    - Forward fetching cursor.
    - Closing cursor after commit or rollback work.
    - Scrollable cursor attribute (static cursor model).
  - Transaction capabilities via the ODBC.INI file and registry:
    - Setting transaction isolation level.
    - Autocommit behavior.

- Statement characteristics:
  - Data-at-execution-time parameters.
  - Asynchronous ODBC API function execution.
- Character set conversions between ROMAN8 and ANSI ISO.

Installation

A suitable product installation method for both client and server will be provided.

For the client side, all existing option entries in the ODBC.INI file will be supported for backward compatibility with the older HP 16-bit ODBC driver. The password stored in the ODBC.INI file may be encrypted for protection.

On the server side, the new listener takes a client's connection request and creates a child process with the client's specified logon. The child process acts upon the requests from the client and terminates upon completion.

Both the older HP listener and the new listener software can run concurrently and take the different connection requests from the old 16-bit clients and the new 32-bit and 16-bit clients.

Bundling

This new PC API is to be bundled at no additional charge with the IMAGE/SQL and ALLBASE/SQL products and will replace the existing 16-bit PC API product.

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.

IMAGE, ODBC, and ALLBASE/SQL

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

SQL for IMAGE was unveiled at the 1991 Interex conference in San Diego. By New Orleans the following year, HP had renamed IMAGE from IMAGE/iX to IMAGE/SQL. Shortly thereafter, at the behest of SIGIMAGE, HP agreed to ship the IMAGE/SQL upgrade to everyone on support in return for a small increment in their support costs. For a few years, HP tracked the percentage of sites who participated in this upgrade program and the last figures I saw showed over 80% participation in this program.

Lots of sites took the plunge knowing that it would be some time before they would make use of the SQL portion of IMAGE. They wanted to get in on the ground floor, Castries, and also show HP management they believed in the direction in which IMAGE was going.

It has now been 4 years since the upgrade program began, 5 years since the first inkling of IMAGE/SQL. Many, if not most of the companies who participated in the IMAGE/SQL upgrade program still have not started to use the SQL portion of IMAGE. There are many reasons for this state of affairs, but chief amongst them is, in one word ALLBASE/SQL.

In 1990, HP pushed hard for existing users to convert applications from IMAGE to ALLBASE. HP believed they could attract more new users to the HP 3000 if it had an RDBMS. That it also ran on the HP 9000 was considered a plus. HP did not count on the grass roots support of the IMAGE users which culminated in the now famous Boston Tea Party. HP, to their credit, immediately reassessed the situation and promised to support IMAGE as well as ALLBASE in the future. The following year, ALLBASE Turbo Connect (ATC) was unveiled. ATC essentially attaches an IMAGE database to an ALLBASE DBE and thus provides SQL access to IMAGE. At first, this was read only (SQL for IMAGE), but quickly encompassed READ/WRITE capabilities (IMAGE/SQL).

The SQL engine technology behind IMAGE/SQL completely resides within ALLBASE/SQL. From a strategic point of view, as well as a resource-allocation and support issue, this made perfect sense. It still makes good sense from those viewpoints but, to a user, it is too complicated. In order to provide SQL access to IMAGE, one has to go through several gyrations and

hoops, some of them flaming. Once this is done, the protocol for supporting the hybrid data storage is unduly taxing at best and nightmarish at worst.

For years, I have been teaching about, talking on, and consulting on installing IMAGE/SQL and getting it ready for client/server access.

The vast majority of users want to use shrink-wrapped tools already on their PCs to get access to the data stored within the IMAGE database. Open Data Base Connectivity (ODBC) is the most readily available middleware which enables one to reach this goal. Most decision support tools, spreadsheet, and databases available on the PC support this technology. All one has to do is enable ODBC access to IMAGE data.

Unfortunately, one must make use of ALLBASE/SQL in order to do this. IMAGE/SQL does not natively support SQL access to the data. No SQL access, no ODBC. No ODBC, no readily available client/server with IMAGE.

Let us look at some of the issues with attaching IMAGE database to ALLBASE/SQL.

- New technology to learn. ALLBASE is not simple like IMAGE. One only needs to look at the pile of books provided by HP to install, manage and maintain ALLBASE to get a flavor for the complexity.
- Creating the DBE is not as simple as one may believe. There are sizing and setting issues which come into play and they cannot be summarily dismissed.
- Attaching the databases may pose problems when one does this across groups and accounts. One may also have to change the IMAGE database schema to account for naming problems.
- Once the databases are attached, the support nightmare begins. If you backup the IMAGE database, you must ensure that the ALLBASE DBE is backed upon the same cycle. If you restore the database or the DBE, the corresponding DBE or database(s) must be restored as well, or at least the extra files created by the attachment (ATCINFO and DBTC files).
- If you make any structural changes in the IMAGE database, you must detach and reattach. Even the automated database tools are not foolproof in this respect and things still fall through the cracks.
- If you copy the attached database, or want to move it into production in another group/account, you are stuck with a detach/attach cycle.
- If you wish to distribute database schemas and software across the enterprise, you are faced with many more steps and synchronization problems.

- One has to periodically synchronize the entry count in IMAGE datasets with the entry count in ALLBASE.
- When users are added to the system, these need to be added to ALLBASE as well, depending on your security scheme.

All these issues, along with others, are a heavy price to pay for simple ODBC access to the IMAGE data. It would be much cleaner and easier, more elegant, if one could totally bypass ALLBASE/SQL. We would then see a whole lot more sites take advantage of IMAGE/SQL and turn the HP 3000 into a server.

HP is over a year late in providing 32 bit ODBC for IMAGE to use on Windows 95/NT and Office 95 products. They are now going to license the M.B. Foster & Associates ODBCLink as the 32 bit ODBC solution.

Besides having had 32 bit ODBC for quite a while already, this smart, far-seeing company also provides the capability to access IMAGE through ODBC without ALLBASE/SQL. Unfortunately, the ODBCLink version which will be given to the sites on support will not include the capability to access IMAGE through ODBC without ALLBASE/SQL.

If you want this capability, you will need to buy it from M. B. Foster & Associates. I do not suggest that MBFA give it away; rather, I am saddened that it was not made part of the deal between HP and MBFA.

IMAGE as a database engine in a client/server architecture will not come to pass for a while, if ever, for most sites if they are forced to use ALLBASE/SQL.

SIGIMAGE Minutes, HP World '96 (Anaheim, California, U.S.A.)

Gary Biggs
Member, SIGIMAGE Executive Committee

SIGIMAGE interim chairman Ken Sletten made a brief introductory statement and distributed the sign-up sheets to record the names of the persons present. Gary Biggs agreed to act as scribe and record the minutes.

The chair opened discussion on when SIGIMAGE should elect officers. Two proposals were presented:

1. To continue electing officers and SIEC members during the HP World Conference, or
2. to move the elections to the spring meeting of SIGIMAGE at IPROF, where it is felt the SIG has more time allotted for such business.

Moving the elections to IPROF was approved by voice vote.

Ken Sletten: new SIGIMAGE Chairman

The second order of business was the election of a new SIGIMAGE Chairman to replace Jerry Fochtman, who stepped aside because of his new position at Bradmark Technologies.

The interim chairman, Ken Sletten, was formally elected to that position by an unanimous acclamation.

32-bit ODBC driver

Jon Bale, the manager responsible for the HP database development team, made a brief presentation announcing the agreement to license M. B. Foster's ODBC connectivity product for distribution as the new 32-bit ODBC driver for the new version of the HP ALLBASE and IMAGE/SQL PC API. Beta testing is scheduled to begin in September with planned distribution by year's end.

B-Trees on IMAGE master datasets

Jon then introduced Bharati Desai, a member of the HP database development team, in preparation for her presentation on B-Trees for IMAGE master datasets.

Bharati presented an excellent, comprehensive look at the implementation and functionality to be introduced with B-Trees on IMAGE master datasets.

Fred White raised several points for clarification on the use of B-trees, and Nick Demos raised a concern regarding the use of this new functionality with older HP language products, such as Transact.

Bharati announced that beta testing for this enhancement was planned for the 4th quarter of 1996, with a planned general release early in 1997.

System Improvement Ballot

The chairman reviewed the status of the top 15 SIG items on the System Improvement Ballot.

The floor was opened for discussion, and minor revisions were made to clarify the top 4 SIGIMAGE enhancement requests.

There were wording changes in the request for 32-bit ODBC to KSAM and flat files, as well as in the request for Adding/Dropping Indices on the Fly.

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
E-mail: membership@interex.org

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
http://www.adager.com