

North American Response Centers

HP 3000 APPLICATION NOTE #16

IMAGE:

CONVERTING TO TURBOIMAGE

and

IMPROVING DATA BASE LOADS



October 15, 1986
Document P/N 5958-5824/2642

HP 3000 APPLICATION NOTES are published by the North American Response Centers twice a month and distributed with the Software Status Bulletin. These notes address topics, where the volume of calls received at the Centers indicates a need for addition to or consolidation of information available through HP support services. You may obtain previous notes (single copies only, please) by returning the attached Reader Comment Sheet listing their numbers.

| <u>Note #</u> | <u>Published</u> | <u>Topic</u> |
|---------------|------------------|--|
| 1 | 2/21/85 | HP 3000 Printer Configuration Guide (superceded by note #4) |
| 2 | 10/15/85 | Terminal Types for HP 3000 HPIB Computers (superceded by note #13) |
| 3 | 4/01/86 | HP 3000 Plotter Configuration Guide |
| 4 | 4/15/86 | HP 3000 Printer Configuration Guide - Revised |
| 5 | 5/01/86 | MPE System Logfile Record Formats |
| 6 | 5/15/86 | HP 3000 Stack Operation |
| 7 | 6/01/86 | Cobol II/3000 Programs: Tracing Illegal Data |
| 8 | 6/15/86 | KSAM Topics: Cobol's Index I/O; File Data Integrity |
| 9 | 7/01/86 | Port Failures, Terminal Hangs, TERMDISM |
| 10 | 7/15/86 | Serial Printers - Configuration, Cabling, Muxes |
| 11 | 8/01/86 | System Configuration or System Table Related Errors |
| 12 | 8/15/86 | Pascal/3000 - Using Dynamic Variables |
| 13 | 9/01/86 | Terminal Types for HP 3000 HPIB Computers - Revised |
| 14 | 9/15/86 | Laser Printers - A Software and Hardware Overview |
| 15 | 10/01/86 | Fortran Language Considerations - A Guide to Common Problems |

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This document contains proprietary information which is protected copyright. All rights are reserved. Permission to copy all or part of this document is granted provided that the copies are not made or distributed for direct commercial advantage; that this copyright notice, and the title of the publication and its date appear; and that notice is given that copying is by permission of Hewlett-Packard Company. To copy otherwise, or to republish, requires prior written consent of Hewlett-Packard Company.

Part 1: Converting to TurboImage

Purpose of this Note

The purpose of this Note is to provide users updating to TurboImage with the information they need to assure a successful installation. The topics to be covered include.

- Descriptions of U-mit and TurboImage
- Installation considerations
- TurboImage resource utilization
- Software and hardware requirements
- Installation and conversion procedures
- Procedure checklist

What is U-mit?

U-mit (MPE version G.02.A0) is the third Base MIT released using MPE V/E internals and table formats. Besides various subsystem updates, this MIT introduces several significant new products:

- TurboImage/3000, the upgraded database management system replacing IMAGE/3000
- TurboImage Profiler, a database applications performance monitor
- LAN/3000 Link, the data communications link for high speed IEEE 802.3 local area networking
- NS/3000, the networking product to provide users extensive capabilities over local area networks

Many other products have also been updated on this MIT and are discussed further in the *U-MIT Communicator*.

U-mit runs on the Series 37, 37XE, 39, 40, 42, 42XP, 44, 48, 58, 64 and 68.

What is TurboImage?

TurboImage is the latest enhancement to the IMAGE/3000 Data Base Management System. Supplied as part of the Fundamental Operating Software (FOS) for the HP 3000, TurboImage provides both improved performance and greater capacity to handle large data base applications. This is done by improving I/O processing, allowing certain data base transactions to operate in parallel, increasing design limits, and providing improved data base recovery and maintenance.

TurboImage was developed in response to customers' needs for larger and more powerful data bases. Advances in HP 3000 hardware and software coupled with increased user sophistication have necessitated the development of this enhancement. TurboImage addresses the requirements of users by providing greater throughput and performance in a proven and reliable data base management system.

Specifically, the changes present in TurboImage include:

- Increased number of data sets and data items per data base
- Increased number of detail records per chain
- Roll-back recovery
- ILR completes last transaction, instead of removing it

- Increased number of concurrent predicate locks
- Improved buffer management, allowing transactions to execute in parallel
- Ability to place data sets on specific devices

Items To Note -- BEFORE Installing TurboImage

DATA BASE CONVERSION. Because of the many new features provided by TurboImage, the format of the root file and master data sets is different from that of IMAGE/3000. Therefore, IMAGE/3000 data bases must be converted to the TurboImage format before they can be used. In general, converted data bases can then be accessed through the normal IMAGE intrinsics and no changes need be made to the application. (See below for more information on program compatibility.)

Since updating to TurboImage includes converting existing data bases, and this conversion can take some time for really large data bases, you may want to schedule the update during a period of low activity. As with any conversion, there is always the possibility that something may go wrong, so you should be very careful to have a good backup completed before beginning. If you have several HP 3000s, you may want to consider installing TurboImage only on one system at first, and testing your applications there.

While existing IMAGE/3000 data bases must be converted before they can be used, they do *not* need to be converted as soon as TurboImage is installed. Therefore, if you have infrequently-accessed data bases, you can convert these as time permits, rather than converting everything at once.

Should you decide to return to IMAGE/3000, the conversion program will convert the data bases back, *as long as* they are still within IMAGE/3000 design limits (65,535 entries per chain, 99 data sets, 255 items, and so forth).

PROGRAM COMPATIBILITY. For the most part, existing IMAGE/3000 programs will run unchanged against TurboImage data bases. However, if the program has some dependency on internal IMAGE/3000 structures or design limits, it may need to be modified to work correctly with TurboImage. For example, IMAGE/3000 has always provided a double-word chain count, but the high-order word was always zero. If a program uses the chain count without looking at both words, it will not operate correctly when the chain count exceeds 65,535. (Since BASIC/3000 does not support a double-integer data type, many BASIC users ignore the high order word of the chain count.)

You should also check to see if any of the following apply to you:

- A new version of Dictionary and Query are supplied with the TurboImage update. Dictionary data bases (DICT.PUB) must be converted using DBCONV, but they need not be reinitialized.
- RAPID products will work with TurboImage provided IMAGE/3000 limits are not exceeded.
- If you have purchased data base transformation tools from a software supplier, you should verify with them that either the software you have will work with TurboImage, or that there is a TurboImage-compatible version available.
- Remote data base access is allowed between IMAGE and TurboImage *provided IMAGE limits are not exceeded*. On a network of systems, you must ensure that all systems which access a TurboImage data base are updated to TurboImage before expanding the number of data items, sets, and items per data set beyond the IMAGE/3000 limits.

TURBOIMAGE RESOURCE UTILIZATION. With its added features, TurboImage uses somewhat more of the HP 3000's resources than did IMAGE/3000.

Because of the complexity of the new buffer management scheme, TurboImage can use more processor time, making it slightly slower for single users and CPU-intensive applications. This primarily is a result of **multi-threading**, a technique which allows multiple users to be accessing TurboImage internal structures simultaneously. (IMAGE/3000's design allowed only one intrinsic call to be in progress against a data base at any given time.) Multi-user applications should see an increase in transaction throughput even with the increased processor utilization, because they will be waiting less for other transactions to complete.

Also as a result of multi-threading, the former Data Base Control Block (DBCBC) has been divided into two structures, the **Data Base Globals (DBG)** and the **Data Base Buffers (DBB)**. The former User Local Control Block is now the **Data Base User (DBU)** and it now contains its own copy of the area used to transfer information between the DBB and the stack. These changes to internal structures cause TurboImage to use slightly more Data Segment Table (DST) entries, and potentially substantially more main and virtual memory, depending on the number of users.

Finally, TurboImage adds one word to the Chain Head for each path controlled by a master data set. (It is the insertion of this word that takes up the majority of the conversion time.) This may cause an increase in the amount of disc space used by master data sets. Specifically, if the new word(s) can be contained in the "wasted" space at the end of the existing blocks, no new disc space will be used. If, however, the block size needs to be increased to hold the longer media record, the new master will use more disc space than the old. How much depends upon the new resultant block size.

PRE-INSTALLATION CHECKLIST. In order to assure the success of the update as well as to provide the ability to return to IMAGE/3000 should this become necessary, you should perform the following prior to installing TurboImage:

- Use DBRECOV to apply any LOG FILES, if necessary.
- Disable ILR, if enabled -- the conversion program will fail otherwise.
- DBCHECK all data bases (broken chains etc.) -- ensure data base structural integrity.
- Restructure data base as needed prior to conversion (capacities, etc.)
- Back up the data base and the schema -- just in case.

Installation Procedure

In order to update from IMAGE/3000 to TurboImage, you must have the appropriate hardware and system software. In most cases, older systems can be upgraded to use the newer version of MPE necessary for TurboImage to run. This section describes the software and hardware requirements of TurboImage, and outlines the installation procedure.

SOFTWARE REQUIREMENTS. TurboImage requires MPE version G.02.A0 (U-mit) or later. If you are on a pre-MPE V/E operating system, we suggest that you consult with Hewlett-Packard prior to updating because of the substantial differences between your software and an MPE-V/E system.

HARDWARE REQUIREMENTS. TurboImage does not have specific hardware requirements. Any system capable of running U-mit can use TurboImage. These on which U-mit can be installed are the HP 3000 Series 37, 37XE, 39, 40, 42, 42XP, 44, 48, 52, 58, 64 and 68. (The Series 70 requires a slightly modified version of U-mit known as U/A-mit, MPE version G.A2.A0. This system also supports TurboImage.)

UPGRADING OLDER SYSTEMS. Certain systems may need upgrades before they can run U-mit.

For the HP 3000 Series 39, 40, 44, and those 42s and 48s without so-called MPE V/E firmware, the upgrade will be necessary in those circumstances where the existing utilization of the Code Segment Table (CST) or of Bank 0 memory is close to the limit. This upgrade consists of a replacement firmware board which allows expansion of the CST beyond 192 entries, and of the other system tables beyond the limits of Bank 0.

For early Series 64 and 68 systems, the Diagnostic Control Unit may need to be replaced before updating to any T- or U-based MIT. If the message `LOADING INIT/IDENT MICROCODE` is displayed after you type `START` or `LOAD`, then you have the old DCU. Your CE can provide the details of the replacement and schedule a time to perform it.

INSTALLATION OF TURBOIMAGE. TurboImage is installed automatically when you update to U-mit; no special action on your part is required. The software is supplied on the FOS tape in your update kit.

Conversion to TurboImage

As described above, after TurboImage is installed, existing IMAGE/3000 data bases must be converted before they can be accessed. The DBCONV utility performs the conversion which involves restructuring the root file and all master sets. Detail data sets remain unchanged. We recommend that you back up data bases prior to converting them with DBCONV.

DISC SPACE REQUIRED. The conversion requires some disc space while it is in progress, and it may also result in a permanent increase in the size of some data sets. When DBCONV runs, it converts the master data sets one at a time by creating a new file of the correct size and then copying the blocks from the old file, inserting the extra word(s) needed to support the increase in the maximum chain length. Once a data set is converted, the old version is purged. Therefore, this temporary extra disc space is equal to the largest master set in the data base. This space is returned when the conversion process completes.

Depending upon the existing blocking of master data set entries, the *permanent* extra disc space requirements could increase by as much as a sector per block, or there may be no increase at all. DBCONV adds whatever sectors are necessary in each case and, in effect, may increase the physical block size. The blocking effectiveness of each master data set is the major factor in determining whether this increase is necessary. In some cases no extra disc space is required to accommodate the added word per path. If you are currently nearing capacity on your discs, you may wish to evaluate the potential increase in disc space for any data bases to be converted to TurboImage.

Before converting the data bases, you can use DBCONV's VERIFY option to analyze the following:

- Total extra disc space required by the converted database.
- Temporary disc space required by DBCONV for conversion of the database.
- The database root file for inconsistencies in data set definitions.

You may :RESTORE DBCONV.PUB.SYS from the U-MIT FOS tape to verify your requirements prior to installing U-MIT itself.

Installation and Conversion Checklist

- 1. Review *U-MIT Communicator*, and the *TurboImage Conversion Guide*.
- 2. Review this Application Note.
- 3. :RESTORE DBCONV.PUB.SYS from the U-MIT FOS tape.
- 4. RUN DBCONV.PUB.SYS,VERIFY against all significant data bases to determine both the disc space needed for the conversion and any permanent disc space increase caused by the conversion. (Remember that the permanent increase will be the *sum* of each data base's increase, while the temporary space required will be the *largest* temporary space shown.)
- 5. Compute the time needed to convert all data bases. This is the sum of the number of sectors occupied by master data sets divided by 1,000,000. The result is in hours. (In other words, DBCONV converts approximately 1,000,000 sectors per hour.) Do not count detail data sets, as they need no conversion.
- 6. RUN FREE5.PUB.SYS to verify that enough space is present to convert your data bases successfully.
- 7. Perform a full backup and check the listing for errors.
- 8. Verify that you completed the previous step.
- 9. Install U-MIT as per the instructions in the U-MIT Software Update Manual which came with your update kit.
- 10. RUN DBCONV.PUB.SYS to convert each database. You may wish to wait to expand data bases to TurboImage limits until after you're sure that you'll stay on U-mit.
- 11. As a precaution, back up the data bases after conversion.

Part 2: Improving Data Base Loads

(Editor's note: the following discussion of IMAGE data base load techniques is also fully applicable to TurboImage users.)

For many applications the IMAGE Data Base Subsystem offers significant advantages over MPE and KSAM files or the filing systems of other operating systems. For this reason conversions from MPE, KSAM or other filing systems to IMAGE Data Bases are common. Because of the potential complexities of IMAGE's internal storage structure however, conversions can be extraordinarily time consuming for MPE users converting multiple data bases or large data bases. This application note describes a method of loading an IMAGE Data Base which can dramatically reduce total load time.

The IMAGE subsystem master key hashing algorithm creates two kinds of master key entries, "primaries" and "secondaries". A "primary" is a key which resides at the exact address that the algorithm created for it. A "secondary" is a key which has hashed to an address that is already occupied by another key, i.e., the "primary" for that address. IMAGE, obviously, cannot put two records at one location and "secondaries" are moved to new locations. A primary and all secondaries are linked together by a set of pointers, thus making a "synonym chain" of all entries that hash to one address. The process of creating secondaries increases overhead because IMAGE must not only search for a new location but must update the chain pointers of the new record and the record which points to it. A chain counter in the primary is also updated.

Things wouldn't be so bad if IMAGE only had to do this once. However, each time a secondary is created IMAGE runs the risk of using an address for it which may become the target of a future primary. When a primary hashes to a location occupied by a secondary, that secondary must be moved and all of the pointers associated with it updated. If the record to be moved is at the end of the chain two records must be updated, the one which is moved and the one which points to it. If the record is anywhere else on the chain three records must be updated, the moved record, the record which points to the moved record, and the record to which the moved record points. E.g., if A points to B points to C, and B is moved pointers in all three records must be updated. The moved record is called a "migrating" secondary.

Now, guess what happens when DBPUT's are done in a tightly packed master which has lots of secondaries. That's right, IMAGE spends an enormous amount of time relocating secondaries and updating pointers. The degree to which load time increases is directly proportional to the frequency with which new keys hash to locations already occupied by secondaries.

Of course, one solution to the problem is to have extremely large master data sets. After all, the more you spread out the data, the less likely a DBPUT will get an address which is already used. Another possible remedy is to choose values for keys which are not likely to hash to the same addresses. These are both important design considerations but they have practical limits.

When these limits are reached an infrequently used IMAGE library procedure, DBGET MODE 8, can be used to speed up a load by eliminating migrating secondaries.

A DBGET MODE 8 returns the entry occupying the primary address of a key value (search argument) even if that entry is not the same key value. E.g., if key A is on the data base at address 123, and key B would also hash to address 123, a DBGET MODE 8 on key B will return key A, regardless of whether or not B is on the data base.

Briefly put, DBGET MODE8 provides a means of checking whether or not a DBPUT for the same key will create a secondary. If a DBGET MODE8 for a given key does not return a record then not only is that record not on the data base, but no other key occupies its address. Thus, a DBPUT can be done for that key without regard for creating or moving (migrating) a secondary. IF a DBGET MODE8 returns a record then a DBPUT for that key will create a secondary.

A "two pass" method of loading a data base which minimizes migrating secondaries in master data sets logically follows. The first pass consists of reading the file to be converted (serially) and doing a DBGET MODE 8 for each record. If no record is found DBPUT is called to add the record to the master set. IF DBGET MODE 8 returns a record then the new record is written out to an MPE file; it is not written to the data base. This is repeated until all records in the "conversion" file have been read. At this point all records which have been added to the data base are primaries. Any other record added to the data base will not hash to a location occupied by a secondary. Thus the problem of migrating secondaries is eliminated for subsequent DBPUTS to the data base. The second pass consists of reading the MPE file containing records not added during the first pass and doing a DBPUT for each record. During the second pass each DBPUT will create a secondary because each record will hash to an address already occupied by a primary. Because all keys hash to an occupied primary address (and not a secondary address) there will be no migrating secondaries.

Although part of the conversion file must be read twice the total time required to load the data base can decrease dramatically because the overhead involved in migrating secondaries is completely eliminated.

