~~Att~~ DTM

North American Response Centers

HP 3000 APPLICATION NOTE #25

TurboIMAGE: Transaction Logging

HEWLETT PACKARD

April 1, 1987
Document P/N 5958-5824R2715

HD9

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.*

Note #	*Published*	*Topic*
1	2/21/85	*HP 3000 Printer Configuration Guide (superseded by note #4)*
2	10/15/85	*Terminal Types for HP 3000 HPIB Computers (superseded by note #13)*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, TERMDSM*
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*
16	10/15/86	*IMAGE: Updating to TurboIMAGE & Improving Data Base Loads*
17	11/01/86	*Optimizing VPLUS Utilization*
18	11/15/86	*The Case of the Suspect Track for 792X Disc Drives*
19	12/01/86	*Stack Overflows: Causes & Cures for COBOL II Programs*
20	1/01/87	*Output Spooling*
21	1/15/87	*COBOLII and MPE Intrinsics*
22	2/15/87	*Asynchronous Modems*
23	3/01/87	*VFC Files*
24	3/15/79	*Private Volumes*

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.

TurboIMAGE: Transaction Logging

INTRODUCTION

1. What is Transaction Logging?

Transaction Logging is a feature of the TurboIMAGE Data Base Management System that allows all changes to a TurboIMAGE data base to be written to a disc or tape file. These changes can be reapplied to the data base in the event of a system interrupt. If specified, a set of data base changes can be treated as one transaction; that is, they can be applied together or removed (rolled out) together.

This discussion was intended for TurboIMAGE users, however, much of it applies to IMAGE/3000 users as well. Information that is different for the IMAGE/3000 Data Base Management System has been marked with an "*" and footnoted.

2. What is a Transaction?

A transaction is the minimum unit of data base change that will leave the data base logically intact. This is either a single DBPUT, DBUPDATE or DBDELETE call, or a group of such calls bracketed by a call to DBBEGIN and DBEND. An example would be that of changing the customer number of a customer record in a master set. The customer number is, most likely, a key and cannot be modified. Moreover, it is likely that there are orders associated with this customer. To change this customer's customer number it is necessary to:

1. *DBDELETE* any associated detail records
2. *DBDELETE* the customer record
3. *DBPUT* the customer record with the new key value
4. *DBPUT* the associated details

If this chain of events is interrupted, the data base is logically inconsistent; therefore, all the above steps must be treated as a single atomic modification, or a transaction.

3. Why do Transaction Logging?

Transaction Logging is intended to bridge the gap between daily backups. Any time after the first users get into the data base after a backup, the changes they make to the data base are subject to loss due to unforeseen events such as power failures and soft or hard system failures. Soft failures are caused by events such as system interruptions or abnormal terminations of TurboIMAGE's DBPUT or DBDELETE intrinsics. Hard failures are due to events such as disc head crashes or media failures. Unless it is more cost-effective for your users to re-enter their work manually, Transaction Logging can save your company time and money.

Transaction logging used in conjuction with user-defined transactions will ensure that a logically consistent database can be realized in the event a soft or hard failure occurs. This combination provides for the most comprehensive form of data base recovery in that it lessens the amount of data loss and manual rework.

Transaction Logging without user-defined transactions will offer protection from excessive data loss and rework in the event a hard failure occurs. After recovery, the data base will be physically consistent but extra effort will be required to analyze what must be done to make the data base logically consistent.

Another form of protection is Intrinsic Level Recovery (ILR). ILR used alone will ensure a physically consistent data base after a soft failure. ILR does not provide logical integrity nor protection from data loss due to hard failures.

The combination of transaction logging without user-defined transactions and ILR will provide protection from excessive data loss due to hard failures. The combination of transaction logging with user-defined transactions and ILR is only required if you wish to use TurboIMAGE's Rollback Recovery Feature.*

4. The "Players"

The three separate software entities that contribute to Transaction Logging include:

- *MPE User Logging.* This is part of MPE, and is available through the MPE commands (:GETLOG, :LOG, :CHANGELOG, etc.) and through six intrinsics (OPENLOG, WRITELOG, BEGINLOG, ENDLOG, FLUSHLOG and CLOSELOG).

- *TurboIMAGE.* TurboIMAGE uses MPE User Logging by calling the logging intrinsics from TurboIMAGE code. To MPE User Logging, TurboIMAGE looks just like any other application that might make use of it.

- *Your Application.* Your programs need not change to make use of logging , because each DBPUT, DBDELETE and DBUPDATE is treated as a transaction. You will, however, need to add calls to DBBEGIN and DBEND if transactions in your application span several changes on the data base.

5. Roll-forward vs. Roll-back *

Roll-forward and roll-back are the two main Transaction Logging types. Roll-forward involves restoring the data base from a backup and reapplying all subsequent changes, as recorded in the log file. Roll-back means backing out those transactions that were incomplete at recovery time. Roll-forward starts from the backup and comes forward to the present; roll-back starts with the present and moves backwards.

*This feature is available with TurboIMAGE only. Image/3000 does not support the roll-back recovery feature, only roll-forward recovery is possible.

Faced with a choice of reapplying 20,000 data base changes versus backing out the changes that might have been incomplete at the time of the failure, roll-back would seem to be the more attractive choice. However, roll-back requires *Intrinsic Level Recovery* (ILR) to be enabled; if you are not familiar with ILR see the *TurboIMAGE Reference Manual* (Part No. 32215-90050) section7. This can be a disadvantage, since ILR adds the additional overhead of two* disc I/O's per DBPUT or DBDELETE. Additionally, with roll-back you are starting with the present state of affairs, which could involve data base damage. The reason roll-forward does not require ILR is that you first restore the data base, thereby eliminating any loss of integrity that the data base might have suffered because of the system interrupt.

In summary, the more secure the data base must be, the more overhead must be incurred as a cost. That cost can be paid at recovery time (roll-forward) or log time (roll-back).

6. System Configuration

Some MPE system configuration values may have an impact on Transaction Logging on your system. The two configurable values are: dialog step 95, MAXIMUM NUMBER OF USER LOGGING PROCESSES, and dialog step 96, MAXIMUM NUMBER OF USERS PER LOGGING PROCESS. These can be found in the *MPE V System Operation and Resource Management Reference Manual* (Part No. 32033-90005) section 7, in the '*SYSTEM TABLE CHANGES?*' dialogue.

Step 95: MAX NUMBER OF USER LOGGING PROCESSES = xxx?

THE MAXIMUM NUMBER OF USER LOGGING PROCESSES, limits the number of :GETLOGs and subsequent :LOGs you can do on the system. Each :GETLOG and associated :LOG command executed for a different log ID identifies a unique logging process. This value is required because the system needs to anticipate how large to build the User Logging Table. This configurable value can be as large as 64. Note that a reload is needed to change this value.

Step 96: MAX NUMBER OF USERS PER LOGGING PROCESS = xxx?

The second value, THE MAXIMUM NUMBER OF USERS PER LOGGING PROCESS, determines how many processes can be logging via any one log process (determined by log ID) and therefore able to access a single log file. Note that this figure does not represent the number of people running the program. If someone opens the data base in read-only mode, that person does *no* logging. If a user program does two DBOPENs of a data base, both read/write, then this user counts as *two* users, for the purposes of MPE User Logging. This number can not exceed 128 on the E/F.00.00 release of MPE V. The maximum number of users per logging process on the G.00.00 release of MPE V has been expanded to 256.

7. Logging to Tape vs. Disc

There are arguments for each. On disc, should you have a head crash, the integrity of the log file may be no better than that of your data base. If you reach EOF on disc and the automatic CHANGELOG (;AUTO parameter of :GETLOG)** has not been enabled, then logging stops, subsequent transactions are rejected and data may be inconsistent. On tape, the tape drive is dedicated to logging for the entire day; transactions go to disc temporarily to smooth data flow. If end-of-tape is reached, transactions stay on

*IMAGE/3000 ILR adds the additional overhead of *three* disc I/O's.

**This feature is available with TurboIMAGE only.

disc until the operator mounts another reel. The operator must be careful to mount a new tape and not just place the drive on-line. Otherwise, MPE will continue logging with the same tape, destroying the existing contents. If you have frequent parity errors on your tape drives, or problems reading tapes that have been written, then your log file may be safer on disc even if you aren't able to segregate your log file on a different disc drive from your data base.

The following summarizes the benefits and drawbacks of logging to tape and of logging to disc:

Logging to tape:

- does not take up disc space (except for the disc buffer).
- is more secure in terms of a hard crash.
- has similar overhead to logging to disc.
- requires a dedicated tape drive.
- requires a reliable tape drive and library of "good" tapes.
- is more time consuming. After a system failure MPE must rewind the tape and scan it sequentially until the EOF is detected. Remaining records are then appended to the file.
- is subject to delays if the console operator is not ready to mount the next reel right away.
- does not provide any security measures to prevent overwriting the current tape.
- is vulnerable to loss of data in the event that a WARMSTART is not done after a system interrupt.

Logging to disc:

- does not require a WARMSTART after a system interrupt.
- has similar overhead to logging to tape.
- is susceptible to a hard crash. This is particularly dangerous if the log file is on the same disc as the data base; a single head crash may destroy both the data base and the log file. If at all possible, it is advisable to put these separate operations on different disc drives. (NOTE: the ;AUTO option may *not* build the new log file on the same disc as the old.)
- is vulnerable to filling the disc file if the automatic CHANGELOG* is not enabled. If EOF is reached, logging will stop and applications requiring logging services will incur an exceptional error and possibly terminate.

*This feature is available with TurboIMAGE only.

8. Logging multiple data bases to one log file

Transaction Logging provides the ability to log more than one data base to one log file. Currently, a maximum of 128 or 256 user processes (depending on the version of MPE) may share a log file at one time. Note that read-only users do not count, and multiple DBOPENs count additionally. Each call to DBOPEN logs out the fully-qualified data base name. Therefore, data bases with the same name and different group names may be logged to the same log file and distinguished at recovery time.

9. Private Volumes and Serial Disc

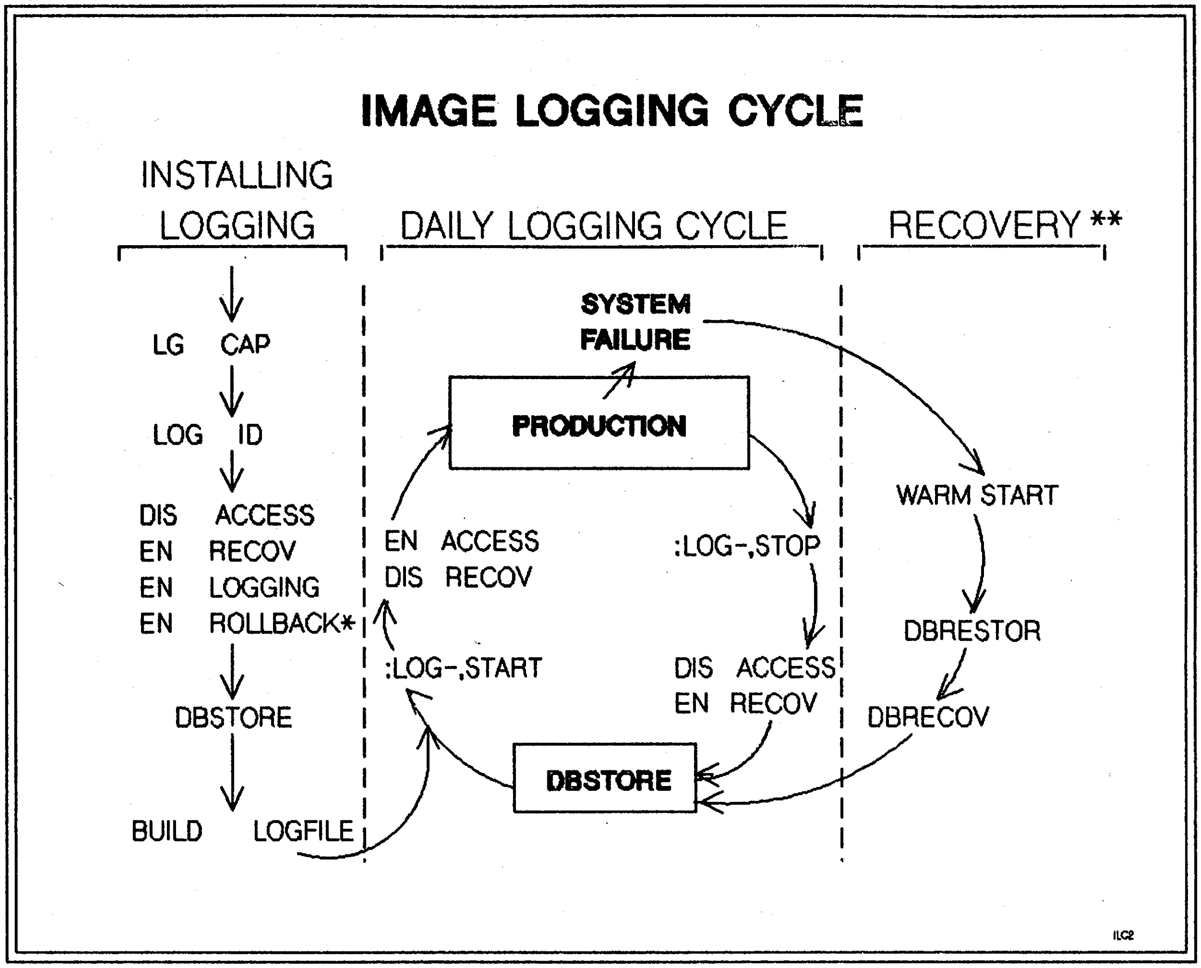
Logging can be done to private volumes. There may be a slight performance degradation over logging to system-domain discs, because the MPE File System now has to access two directories. The use of private volumes can enhance data integrity. Since you can mandate that the data base and log file be on separate private volumes, or one in the system domain and one on a private volume, you can avoid the possibility of one head crash destroying both. The one detail to keep in mind is that you need to sign on to the group and account where the log file resides to issue the :GETLOG command.

Logging to serial disc and cartridge tape is also allowed. If logging to cartridge tape, the device *must* be in the device class CTAPE. If logging to serial disc, the device *must* be in the device class SDISC.

10. The Three Phases of Logging

The typical site experiences three phases of Transaction Logging. 1) set up 2) the daily logging cycle, and 3) recovery of applications from a system interrupt using the log files. This note will look at the mechanics of logging through these three phases.

If you are not familiar with the various aspects of Transaction Logging, you may want to read sections 7 and 8 in the *TurboIMAGE Reference Manual* before continuing.



TurboIMAGE Logging Cycle, figure 1

EN = enable
DIS= disable
*This feature is available with TurboIMAGE only.
**Applies to roll-forward recovery only.

BEFORE THE LOGGING CYCLE: SETTING UP LOGGING

To set up logging, five steps are taken:

Step 1: Acquire LG (Logging) Capability

Step 2: Acquire Log ID

Step 3: Set Log ID and Flags into Database

Step 4: DBSTORE the Data Base

Step 5: Build the Log File

THE LOGGING CYCLE

This is the logging cycle itself, the second of the three phases. Consider this an "infinite loop"; from now until the end of your application's life your data processing shop will execute this loop on a regular basis. When a system interrupt occurs, you will "branch" out of this loop and drop down to the third phase, recovery.

The logging cycle consists of six steps:

Step 1: Start the Log Process

This is by means of the :LOG command, as documented in the *MPE V Commands Reference Manual* (Part No. 32033-90006) and the *TurboIMAGE Reference Manual.*

If you are logging to tape, be aware that *MPE User Logging*, which actually requests the tape and writes to it, will use the Labelled Tape Facility. For your information, if you are not familiar with the use of labelled tapes on the HP 3000, two operator requests will appear on the console. The first one will look like:

```
9:45/61/Volume ID for TOADLOG (MAX CHARS=6)?
```

What is a volume ID? That is the label on the labelled tape. When you mount an unlabelled tape, a message appears on the console that says `VOLUME (UNLABELLED) MOUNTED ON LDEV` *nnn.* In the case of a labelled tape, it will give the six-character (or less) volume ID instead of "(unlabelled)". It may be useful for you to remember that, because if you should forget the volume ID you replied with, you can always mount the tape on the tape drive, and the system itself will give you the volume ID.

The second operator request will look like:

```
9:45/61/Mount tape of volumeset TOAD (ANS)
```

The expected reply is the ldev number of the tape drive to be used for logging.

Step 2: Set Flags

Check the *TurboIMAGE Reference Manual* for the details about these flags. Also, note the discussion about these flags below, in step 5.

Step 3: Production

This is where you let users access and modify the data base. If a system interrupt of any sort occurs between now and step 6 below, we must "branch" out of this loop and go to the Recovery Phase, see the following section titled *When the Logging Cycle is Interrupted.*

Step 4: Stop Log Process

Step 5: Set Flags

The two flags are set via DBUTIL and are documented in the *TurboIMAGE Reference Manual* . Note that this step is not strictly required at this point; the rationale is that you are setting the flags the way you want them on the backup, because at DBRESTOR and DBRECOV time this is an extra step that could be easily overlooked.

Step 6: DBSTORE the Data Base

A DBSTORE is recommended here. If you use a MPE :STORE, you are circumventing a check that was built into the Transaction Logging system. By doing so you introduce the risk of human error since you have to depend on your own records to ensure that the :STORE was done and that you have the right tape set. This is because when DBRECOV is run, it checks a flag in the root file to determine when the last DBSTORE was done. There is a CONTROL option in DBRECOV (CONTROL NOSTORE) that tells DBRECOV to ignore the fact that a DBSTORE wasn't done.

"What if..."

Suppose you do a :LOG logid, RESTART instead of a :LOG logid, START after the DBSTORE. What does this do to the recovery process? The major difference between START and RESTART is the treatment of the log file. START causes a new log file to be used, and the sequence numbers of the records in it begin with 1. RESTART resumes logging to the same log file, and the sequence numbers will pick up where they left off elsewhere in the file. Note that the "file" may be several physical files, but to DBRECOV they are one logical "file", and recovery of this logical file may begin only at one place, the beginning. DBRECOV will not begin recovery from a physical file that does not begin with sequence number 1.

Suppose you do a :CHANGELOG* at some point. You'll get a new disc or tape file, with the three digits on the end of the file name incremented by one. If a system interrupt were to occur after this point, could you begin recovery starting at this new file? No, for the same reason as above. Though physically a separate file, this is treated as part of one logical file by DBRECOV; sequence numbers in this physical file do not begin with 1.

WHEN THE LOGGING CYCLE IS INTERRUPTED: RECOVERY

There has been some sort of system interrupt, and now you must run recovery on your data bases. You will need to follow one of the following sequences of steps, depending on whether you are logging to disc or to tape. Note that the intent here is to provide an overview of the steps that need to be taken and to supplement the information in the manual.

If logging to tape:

Step 1: WARMSTART the System

When you log to tape, data is buffered in a file in PUB.SYS named ULOGnnnn, where nnnn is four digits starting with 0001. This file is flushed to tape periodically as it fills up. When the system fails and recovery is required, chances are you have some records sitting around in that buffer; you must get them to tape before you can recover with that tape. This is the function of the "MPE Cleanup Mode"

*This feature is available with TurboIMAGE only mentioned in the *TurboIMAGE Reference Manual*, Section 7. If you are logging to tape, a WARMSTART is essential to the logging process.

Step 2: Reply to the Tape Request

This is the tape request for the operation described above.

Step 3 (if using roll-forward recovery): Purge and Restore the Data Base

Step 3 (if using roll-back recovery): :STORE the data base

If you are doing roll-back recovery, you won't be using your backup, since you'll be starting with the data base as it currently exists on disc and rolling data out. A backup of the *current* data base is recommended should a problem occur during recovery.

Step 4: Set Flags

Step 5: Set File Equation

Remember the discussion about labelled tapes? You need to issue a file equation for the tape:

```
:FILE logfilename;DEV=TAPE;LABEL=volid
```

where "logfilename" is the name of the log file as entered in the :GETLOG command and displayed by the :LISTLOG command, and "volid" is that max-6-character volume ID entered as a reply to a console request. Note: the *TurboIMAGE Reference Manual* includes the ;LABEL= in the example of the STATS parameter in the DBRECOV documentation, but the *IMAGE/3000 Reference Manual* (Part No. 32215-90003) does not include this example.

Step 6 (for multi-reel log files only): Mount the first reel of the set

If your log file spans more than one reel, you'll now need to mount the first reel of the set. DBRECOV will need to scan the logfile to locate roll-back candidates.

Step 7: Run DBRECOV

Roll Forward - use RECOVER command

Roll Back - use ROLLBACK command

If logging to disc:

Step 1: WARMSTART the System

A WARMSTART is not strictly necessary for logging to disc, but you may have spoolfiles on the system you need to recover, and only a WARMSTART will retain the spoolfiles on the system.

Step 2 (if using roll-forward recovery): Purge and Restore the Data Base

Step 2 (if using roll-back recovery): :STORE the data base

If you are doing roll-back recovery, you won't be using your backup, since you'll be starting with the data base as it currently exists on disc and rolling data out. A backup of the *current* data base is recommended should a problem occur during recovery.

Step 3: Set Flags

Step 4: Set File Equation

As with tapes, a file equation for the log file is required, so that DBRECOV can find it on disc. It will look like:

```
:FILE logfilename;DEV=DISC
```

where "logfilename" is the name of the log file as entered in the :GETLOG command and displayed by the :LISTLOG command.

Step 5: Run DBRECOV

QUESTIONS & ANSWERS

Following are some common questions and answers about Transaction Logging. Some are reprinted from past *Response Center Questions and Answers*; some are new.

Q. *The TurboIMAGE Reference Manual* states that TurboIMAGE does not provide a mechanism for recovering a transaction that spans multiple data bases; so how do I go about providing that ability in my application?

A. There's no "right" answer to this question, but here's one suggestion. Build a separate data set or data base to be used for logging purposes. During logging, follow a multiple-data-base transaction with a DBPUT of a record to this set or base. The purpose of this record would be to declare that transaction *nnnn* is complete. The user recovery program would need to determine whether or not the transaction is complete by checking for the presence or absence of this record.

Q. If a system failure occurs, can I start logging and data base modifications without performing a recovery first?

A. *No* , unless you are sure that the data base was *not* open for update access at the time of the failure. Otherwise, it is necessary to run DBRECOV and follow it with one of the three post-recovery options (*TurboIMAGE Reference Manual*, Post Recovery Procedures Section).

Q. What happens if a disc logfile runs out of space?

A. That depends. TurboIMAGE offers the capability of automatically switching log files as they get full, but it requires some preparation. The :GETLOG command contains special syntax (the ";AUTO" keyword); see the *MPE V Commands Reference Manual* . In conjunction with this, your log file name must end with "001". Assuming you have fulfilled these requirements, then upon filling the file the system will close it, build a new file after incrementing the trailing three digits of the file name ("001" -> "002") and allow logging to resume. Otherwise, the logging process will terminate and TurboIMAGE intrinsic calls will return errors.*

Q. What happens when a tape log file reaches the end-of-tape?

A. A message will be issued to the system console to mount another tape. Normal user activity will continue, provided the tape is mounted in a timely fashion. If not, the disc buffer file will fill up and the user programs will suspend. Normally, the operator has several minutes before there is danger of the disc buffer filling.

*IMAGE/3000 doesn't have the capability of automatically switching log files. The logging process will terminate and IMAGE/3000 intrinsics will return errors.

Q. What steps are necessary to recover a data base after a system failure?

A. 1. WARMSTART the system. 2. If logging to tape, reply to the logging process tape request. The operation being performed is that of flushing the disc buffer file to tape. 3. If doing roll-forward recovery, purge and restore the backup copy of the data base. 4. If roll-forward, Use DBUTIL to verify the status of the following flags:

```
ACCESS   is  DISABLED
RECOVERY is  ENABLED
LOGGING  is  ENABLED
```

5. Set file equations for log filename.

6. Run DBRECOV to recover the data base.

Q. Does Transaction Logging log BEFORE or AFTER the data base is actually updated?

A. BEFORE. The WRITELOG intrinsic is used to log information when the TurboIMAGE intrinsics DBPUT, DBDELETE and DBUPDATE are called. WRITELOG is called after all the TurboIMAGE error checks are made, but before the actual disc modifications are made to the data base. Consequently, a log record is not written until the TurboIMAGE procedure has committed itself to succeed, with the exception of file system or similar failures.

Q. What happens with logging if disc caching is enabled and the system fails? Do any data base modifications get lost? Do any logging records get lost?

A. Possibly. It is recommended that the :CACHECONTROL command be used to set BLOCKONWRITE = YES. This will cause the process requesting the data modification to pause until the modification is made on the disc. The net effect of BLOCKONWRITE = YES is this: for disc writes (this includes DBPUT, DBDELETE and DBUPDATE) it is as if caching wasn't turned on. For disc reads you still have the performance gain of caching, and for disc writes you still have the safety of data integrity but lose caching's performance gains.

Q. Can the log file be accessed by user-written programs? If so, what do the record layouts look like?

A. Yes, it can. The record layouts are in Appendices E and F of the *TurboIMAGE Reference Manual* and *IMAGE/3000 Reference Manual.* Keep in mind that Transaction Logging is merely an application to MPE. Glance at the format for the WRITELOG record, on p. F-1 of the *TurboIMAGE Reference Manual* ; the area following word 8, the "user buffer area", is the TurboIMAGE information described in Appendix E. A TurboIMAGE log record will contain 8 words of MPE overhead and then the TurboIMAGE overhead and user data.

Q. Is there a minumum amount of Virtual Memory I must have on ldev 1? If so, how much and why?

A. According to the *System Operation and Resource Management Manual* (Part No. 32033-90005) if you are on MPE Version G.00.00 (MPE V/E) or later, you need to configure at least 4 kilosectors of virtual memory on logical device 1. Although the SYSDUMP dialogue will allow you to configure as little as 1 kilosector, this is not recommended for most systems. In most cases, configuring 1 kilosector will only work if sufficient virtual memory space is allocated on another system domain disc.

Part of the reason for the recommended minimum of 4 kilosectors is that the Job Master Table (JMAT), Input Device Directory (IDD), and Output Device Directory (ODD) are stored in virtual memory on ldev 1. These tables are necessary for the WARMSTART system startup option. During a WARMSTART these tables are *not* reconstructed. Instead, they are recovered so jobs with the ;RESTART parameter on the job card will be reintroduced and input/output spoolfiles will be recovered.

Although you need a minimum of virtual memory on ldev 1, it is recommended that you do not configure all virtual memory on that drive. This is because virtual memory allocation on ldev 1 can only be changed during a RELOAD. Also, by allocating all but the minimum requirements for virtual memory on another system domain disc, ldev 1 (which is already used extensively since it contains the system library, directory, etc.) may have better performance. VM changes on discs besides ldev 1 can only be made when restarting the system during INITIAL. This can not be accomplished by creating a sysdump tape with VM changes on it.

Guidelines for determining virtual memory space can be found in Chapter 5 of the *System Operation and Resource Management Manual* under "Altering the Size of Virtual Memory" in "The System Disc" section. Also, if you have OPT/3000 you can use it to determine if your current virtual memory allocations are sufficient.

Q. Why do I occasionally get a LOAD WARN 89, 90, or 91 (PROGRAM LOADED WITH LIB=S, PROGRAM LOADED WITH LIB=P, or PROGRAM LOADED WITH LIB=G) when I try to run a program?

A. This happens when two persons run a program at the same time and they use different LIB= qualifiers. For example, suppose you ran a program called "XYZ" using the command :RUN XYZ;LIB=P and you received the message:

```
PROGRAM LOADED WITH LIB=G (LOAD WARN 91).
```

In this example, someone was already running the program, and that person had run the program using the ;LIB=G parameter. You might have also received the message:

```
PROGRAM LOADED WITH LIB=S (LOAD WARN 89)
```

This would mean the first person either didn't specify a LIB= at all or used LIB=S on the :RUN command. In either case, the message is meant to be informational only; letting you know that your execution of the program will be using a different library than you specified.

more...

Q. What does MPE require in a system's I/O configuration? I could really use some clarification on this point.

A. Although most sites follow certain conventions when configuring their systems, in most cases MPE does not require specific I/O configurations. The absolute minimum configuration for any system is:

a. The system disc (configured as LDEV 1).
b. The LOAD device (1/2" tape, cartridge tape, serial disc). This can have any LDEV number.
c. The system console (configured as DRT 8, unit 0). This too can have any LDEV number.
d. Class DISC must appear on at least one system-domain disc.
e. Class SPOOL must appear on at least one system-domain disc, if spooling is to be used.

This configuration, while it will work, is not very usable because there is no line printer, no STREAMs device, and no DDUMP device. But, strictly speaking, none of these is required.

Here are some common misconceptions regarding required I/O devices. Note that each of the following initial statements is *not true*.

1. My STREAMs device must be LDEV 10.

   The only requirement for the STREAMs device is that it be some job-accepting device other than a terminal. In fact, when card readers were common on the HP 3000, the STREAMs device was usually the reader, which by convention was configured as LDEV 5.

2. My STREAMs device must have class JOBTAPE on it.

   The STREAMs device can have *any* class, or none at all.

3. The system console must be LDEV 20.

   The console must be on DRT 8, unit 0; however, it can have any LDEV number you want (except 1, since that is the system disc).

4. The system disc, LDEV 1, must have class SYSDISC.

   There is *no* requirement that LDEV 1 have a specific class name. There must be *some* system-domain (non PV) disc in class DISC (and class SPOOL if you're spooling), but that's the only requirement.

5. Private Volumes cannot have class DISC.

   They can, although this won't meet the requirement for having one system-domain disc in class DISC.

In summary, the items listed in the "minimum configuration" list above are the only ones MPE absolutely *requires* in the I/O configuration. Anything else that you see is done that way *by convention*, and may be different from one system to the next with no ill effects. However, you may find that it is easier for you and others to refer to and service your system when you follow a conventional configuration.

READER COMMENT SHEET

North American Response Centers
HP 3000 Application Note #25: TurboIMAGE: Logging
RC Questions & Answers (April 1, 1987)

We welcome your evaluation of this Application Note and attached RC Questions & Answers Sheet. Your comments and suggestions help us to improve our publications. Please explain your answers under Comments, below, and use additional pages if necessary.

	AppNote	RC Q&A
Is this publication applicable to your site?	□ Yes □ No	□ Yes □ No
Are the concepts and wording easy to understand?	□ Yes □ No	□ Yes □ No
Would you like to see additional Notes on this subject?	□ Yes □ No	□ Yes □ No

Comments and/or suggestions for future Application Notes:

This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

FROM: **Date** ___________

Name ______________________________

Company ______________________________

Address ______________________________

____

____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1710, SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Application Note / RC Q & A Comments
Hewlett-Packard Western Response Center
3300 Scott Boulevard
Santa Clara, CA 95054

FOLD FOLD