Worldwide Response Center

HP 3000 APPLICATION NOTE #41

MPE FILE LABEL REVEALED

HEWLETT PACKARD

REPRINT June 15, 1988
Document P/N 5958-5824R2827

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

RESPONSE CENTER APPLICATION NOTES

HP 3000 APPLICATION NOTES *are published by the Worldwide Response Center twice a month and are distributed with the Software Status Bulletin. These notes address topics where the volume of calls received at the Center indicates a need for addition to or consolidation of information available through HP support services.*

Following this publication you will find a list of previously published notes and a Reader Comment Sheet. You may use the Reader Comment Sheet to comment on the note, suggest improvements or future topics, or to order back issues. We encourage you to return this form; we'd like to hear from you.

MPE FILE LABEL REVEALED

MPE files are composed of a system file label, possible user labels, and data records. The intent of this note is to explain these file labels:

- How can you read file labels?
- How can you use this information?
- What can you do if they go wrong?

The first section of this note will lay out the file label. In the next section, the tools available for examining the file label and the various formats will be discussed. This is followed by a section discussing, in detail, each part of the label with an example. Following this section, is a discussion of the problems that might indicate there is a problem with the file label. The last section is on the various utilities to deal with diagnoses and resolution of file labels identified as causing a problem.

A. FILE LABEL SCHEMATIC

The file label occupies the first sector of the first extent occupied by the file. This is followed by user label(s) on the next sector boundary, but in the same block. Following the user label(s) in the next block is the actual data.

Let's look at the structure of the file label. The layout for this label is contained in the MPE V COMMANDS MANUAL (P/N 32033-90006) and is included here for use later in the discussion of the various parts of the label.

FILE LABEL SCHEMATIC
TABLE A

	Position in octal words	decimal words		contents
1.	0-3	0-3		local file name
	4-7	4-7		group name
	10-13	8-11		account name
	14-17	12-15		identity of file creator
	20-23	16-19		file lockword
2.	24-25	20-21		file security matrix
3.	26	22	(bits 0:7)	reserved for Native Language Support
			(bits 8:13)	unused
			(bits 14:1)	store/restore release bit
			(bits 15:1)	file secure bit (1 if secure,0 if released)
4.	27	23	(bits 0:7)	file creation date (year of century)
			(bits 7:9)	file creation date (day of year)
5.	30	24	(bits 0:7)	last access date (year of century)
			(bits 7:9)	last access date (day of year)
6.	31	25	(bits 0:7)	last modification date (year of century)
			(bits 7:9)	last modification date (day of year)
7.	32	26		file code (privileged for data sets %177157)
8.	33	27	(bits 0:1)	private volume information - class flag bit
			(bits 1:3)	unused
			(bits 4:4)	mounted volume table index
			(bits 8:8)	volume mask
9.	34	28	(bits 0:1)	store bit (if on, store or restore in progress)
			(bits 1:1)	restore bit (if on, restore in progress)
			(bits 2:1)	load bit (if on, program file is loaded)
			(bits 3:1)	exclusive bit (if on, file is opened exclusively)
			(bits 4:4)	device subtype
			(bits 8:6)	device type
			(bits 14:1)	file is open for write
			(bits 15:1)	file is open for read
10.	35	29	(bits 0:8)	number of user labels written
			(bits 8:8)	number of user labels
11.	36-37	30-31		maximum number of logical records
12.	40-41	32-33		file control block vector
13.	42	34		checksum
14.	43	35		coldload identity
15.	44	36		foptions specifications
16.	45	37		logical record size (in negative bytes)
17.	46	38		block size (in words)
18.	47	39	(bits 0:8)	sector offset to data
			(bits 8:3)	unused
			(bits 11:5)	number of extents minus 1
19.	50	40		logical size of last extent size in sectors
20.	51	41		extent size
21.	52-53	42-43		end of file pointer for file
22.	54-153	44-107		two word addresses of up to 32 disc extents, beginning with address of first extents (words 44-45)
23.	154-155	108-109		file restore time
24.	156	110	(bits 0:7)	file restore date (year of century)
			(bits 7:9)	file restore date (day of year)
	157	111		unused
25.	160-161	112-113		start of file block number
26.	162-163	114-115		block number of last block
27.	164-165	116-117		number of open and closed records
28.	174-177	124-127		device class

B. TOOLS FOR LOOKING AT LABELS

There are several different ways to look at the label or parts of labels. The command :LISTF and the utility LISTDIR were created specifically to allow users to look at label information in a usable format. The following table describes the capabilities necessary to run these commands.

CAPABILITIES NEEDED FOR LISTING FILE LABELS

TABLE B

COMMAND	GROUP FILES	ACCOUNT FILES	ANY FILE ON SYSTEM
Listf (,0)	standard user	standard user	standard user
Listf,1	standard user	standard user	standard user
Listf,2	standard user	standard user	standard user
Listf,-1	account manager	account manager	system manager
Listdir(listf)	standard user	standard user	standard user
Listdir(listf;pass)	creator	account manager	system manager
Listdir(listf;map)	creator	account manager	system manager
Listdir(listsec)	standard user	standard user	standard user

WHAT EACH COMMAND WILL SHOW ABOUT A FILE LABEL

TABLE C

Each of these commands gives a different perspective of the file label. Using the file label layout, here is a description of what each of these commands will display. Note the :LISTF,0 command simply lists the name of the file.

FIELD DESCRIPTION	Listf,1	Listf,2	Listdir Listf,-1	Listdir Listf	Listdir Pass	Listdir Listsec
local file name	x	x	x	x	x	x
group name	x	x	x	x	x	x
account name	x	x	x	x	x	x
identity of file creator			x		x	
file lockword			x		x	
file security matrix			x		x	x
Native Language Support			x			
Store/restore release bit			x			
File secure bit			x			
File creation date (year)			x	x	x	
File creation date (day)			x	x	x	
Last access date (year)			x	x	x	
Last access date (day)			x	x	x	
Last modification date(yr)			x	x	x	
Last modification date(day)			x	x	x	
File code			x	x	x	
Private volume information			x	x	x	
Mounted volume table index			x			
Volume mask			x			
Store bit			x	x	x	
Restore bit			x	x	x	
Load bit			x	x	x	
Exclusive bit			x	x	x	
Device subtype			x	x	x	
Device type			x	x	x	
File is open for write	x	x	x	x	x	
File is open for read	x	x	x	x	x	
Number of user labels			x	x	x	
Maximum number of records	x	x	x	x	x	
File control block vector			x	x	x	
Checksum			x			
Coldload identity			x	x	x	
Foptions specifications			x	x	x	
Logical record size	x	x	x	x	x	
Block size (in words)			x	x	x	
Sector offset to data		x	x	x	x	
Number of extents -1		x	x	x	x	
Logical size of last extent			x			
Extent size		x	x	x	x	
End of file pointer	x	x	x	x	x	
Two word addresses of disc		x	x	x	x	
extents beginning with			x			
address of first extent			x			
File restore time			x			
File restore date (year)			x			
File restore date (day)			x			
Start of file block number			x			
Block number of last block			x			
Number of open/closed recs			x			
Device class			x			

C. COMPONENTS OF THE FILE LABEL DISCUSSED

Since the :LISTF command lists the entire contents of the label, we can use this output to dissect a file label.

:LISTF XYZ,-1

The output will look something like this:

```
F  =  XYZ
054131 055040 020040 020040 001437 154123                XYZ........S
054131 055040 020040 020040 050125 041040 020040 020040 XYZ.....PUB.....
051531 051440 020040 020040 046501 047101 043505 051040 SYS.....MANAGER.
020040 020040 020040 020040 020202 004040 000001 127123 ................
127250 127222 177157 000000 005014 000401 000000 000010 .....o..........
000000 000000 145400 023661 000001 176000 001000 002000 ................
000044 000044 000000 000010 001025 040144 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 000000 000000 000000 000000 ................
000000 000000 000000 000000 013404 026400 127157 000000 ................
000000 000000 000000 000000 000000 000000 007023 034411 ..............9.
000000 000000 000000 000000 042111 051503 020040 020040 ........DISC....
```

1. GENERAL INFORMATION ABOUT THIS FILE:

Refer to TABLE A for the label format and you can dissect a label. Using this example you can determine the following about the file:

FIELD NAME	OCTAL CONTENTS	ASCII TRANSLATION
Local file name	054131 055040 020040 020040	XYZ
Group name	050125 041040 020040 020040	PUB
Account name	051531 051440 020040 020040	SYS
File creator	046501 047101 043505 051040	MANAGER
File lockword	020040 020040 020040 020040	<<SPACES>>

2. FILE SECURITY MATRIX:

Octal Value: 020202 004040

```
                    bit numbers              1|1|1 |1|1|1
                    0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
                    ---------------------------------------
Bit map-Word 1 - 0 |0|1|0 |0|0|0 |0|1|0 |0|0|0 |0|1|0
Bit map-Word 2 - 0 |0|0|0 |1|0|0 |0|0|0 |1|0|0 |0|0|0
```

To read this bit map, you ignore bits 0 and 1 of word 1, leaving 30 bits. This makes the bit pattern look like this:

```
10 000 010 000 010 + (word 2) 0 000 100 000 100 000
```

These remaining bits are divided into 5 access categories:

1) read
2) append
3) write
4) lock
5) execute.

When you divide these remaining 30 bits into the five groups you get a bit pattern like this:

```
100000 100000 100000 100000 100000
```

This gives you 6 bits in each of the groups. Each of these bits tells whether or not a user may have the appropriate access to the file. These bits are in the order:

1) any
2) account manager
3) account librarian
4) group user
5) group librarian
6) creator

The matrix looks like this:

```
          ACCESS                    ACCESSOR
          GROUP                       BITS

                                  1 2 3 4 5 6
1     Read access bits      |     1 0 0 0 0 0   -  Any may read
2     Append access bits    |     1 0 0 0 0 0   -  Any may append
3     Write access bits     |     1 0 0 0 0 0   -  Any may write
4     Lock access bits      |     1 0 0 0 0 0   -  Any may lock
5     Execute access bits   |     1 0 0 0 0 0   -  Any may execute
```

3. NATIVE LANGUAGE AND STORE/RESTORE BITS:

Octal Value: 000001

```
            bit numbers              |1|1|1 |1|1|1
            0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   0 |0|0|0 |0|0|0 |0|0|0 |0|0|0 |0|0|1
```

The first eight bits describe the native language attribute of the file. The next-to-last bit is the STORE/RESTORE release bit which designates whether or not the file is being restored. This flag is reset by STORE/RESTORE not by the file system.

The native Language bit is 0/7-00000000; native language is 0 (English).

The STORE/RESTORE release bit which is (14/1) is *off*. This means that STORE/RESTORE is not in progress.

The file secure bit (15/1) is *on*. This means the file is secured (or not released).

4. FILE CREATION DATE:

Octal Value: 127123

```
            bit numbers              |1|1|1 |1|1|1
            0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   1 |0|1|0 |1|1|1 |0|0|1 |0|1|0 |0|1|1
```

The creation year bit 0/6 is 1010111 or decimal 87 (1987).
The creation day (7/15) is 001010011 or decimal 83 (Julian date) or March 24.

The file was created March 24, 1987.

5. FILE LAST ACCESS DATE:

Octal Value: 127250

```
            bit numbers              |1|1|1 |1|1|1
            0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   1 |0|1|0 |1|1|1 |0|1|0 |1|0|1 |0|0|0
```

The access Year bit (0/6) is 1010111 or decimal 87 (1987).
The access date (7/15) is 010101000 or decimal 168 (Julian day 168) or June 15.

The file was last accessed June 15,1987.

6. FILE MODIFICATION DATE:

Octal Value: 127222

```
            bit numbers              |1|1|1 |1|1|1
            0 |1|2|3 |4|5|6| 7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   1 |0|1|0 |1|1|1 |0|1|0 |0|1|0 |0|1|0
```

The modification year bit (0/6) is 1010111 or decimal 87 (1987).
The modification day bit (7/15) is 010010, decimal 146 (Julian day 146), May 26.

This file was last modified on May 26, 1987.

7. FILE CODE:

Octal Value: 177157 or Decimal -401

This is actually a data set from a data base, a privileged file. A file code identifies a special purpose file by the function code contained in this field. A list of these codes and their uses is contained in the MPE V COMMANDS MANUAL (P/N 32033-90006) under the :BUILD command.

8. FILE LABEL PRIVATE VOLUME INFORMATION:

Octal Value: 000000

```
            bit numbers              |1|1|1 |1|1|1
            0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0
```

Class bit: bit (1) 0
Mounted volume table: (2) 0
Volume mask: (4/10) 00000000

This file is not on a private volume. If this file was on a private volume, this bit map would reveal on which volume the label resides. Bit 15 is *on* if it resides on the first volume, bit 14 if it is on the second, and so on.

9. FLAGS:

Octal Value: 005014

```
                bit numbers              |1|1|1 |1|1|1
                0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
                ---------------------------------------
        bit map 0  0 0 0  1 0 1  0 0 0  0 0 1  1 0 0
```

Store bit:	bit (0)	0	Store NOT in progress
Restore bit:	bit (1)	0	Restore NOT in progress (if this bit was set due to a restore in progress the store bit would be set to 1 also and the file would be inaccessible)
Load bit:	bit (2)	0	Program file (if set it means that this file is a loaded program or SL and cannot be modified except for a privileged user. It is reset by the Loader and not the file system)
Exclusive bit:	bit (3)	0	File is NOT not opened exclusively and is possibly available to all users
Device Subtype:	bit (4/7)	1010	Decimal 10 - the device subtype of the first extent of the file is 10
Device type:	bit (8/13)	000011	Decimal 3 - the device type of the first extent of the file is 3
File open for write:	(14)0		File is NOT opened for write access
File open for read:	(15)0		File is NOT opened for read access

This word contains the bits used for locking.

10. USER LABELS:

Octal Value: 000401

```
                bit numbers              |1|1|1 |1|1|1
                0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
                ---------------------------------------
        bit map 0  0 0 0  0 0 0  1 0 0  0 0 0  0 0 1
```

The number of written bits (0/8) is 00000001 or 1 User Label written.
The actual number of User Labels Bits (8/15) is 00000001 or 1 User Label.

User Labels are an area set aside to provide specific identifying information about a file. User labels are stored immediately after the file label on disc. Many HP applications use this label to store information about the file itself. An example of a frequently used application of user labels is data bases. Root files contain multiple user labels and each data set contains a single user label. Data set user labels keep track of information such as the capacity, number of free records available for use, and the delete chain head (the first entry marked as deleted, thus available for reuse when adding to a data set). Another application is the Self Describing File or (SD File) which will hold the information necessary to read the file that follows. Query, for example, uses SD Files for output.

11. NUMBER OF LOGICAL RECS:

Octal Value: 000000 000010 or Decimal 8

This is a double word integer representing the maximum number of records allowed for this file.

12. FILE CONTROL BLOCK VECTOR:

Octal Value: 000000 000000

If this number is 0, the file is not being accessed. However, if it is greater than 0, it contains the offset to the file control block. The file control block is an extra data segment that allows coordination of access to the file while it is on a sharable device, such as a disc. This data segment contains information from the file label and might exist in the user's stack, if the file is opened exclusively; or in main memory, if the file is shared. The advantage to this is faster access.

13. CHECKSUM:

Octal Value: 145400

This number is used for error detection. Every time the file label is read from disc this is calculated using a standard algorithm. It is then compared against the value stored in this field. Then every time the file label is written it is recalculated according to this algorithm and inserted into this position.

14. COLDLOAD IDENTITY:

Octal Value: 023661

This is the coldload number in effect the last time this file was accessed. It should be the current coldload identity number. If not, it means the system crashed while the file was opened and some data in this file needs to be reset.

15. FOPTIONS SPECIFICATIONS:

Octal Value: 000001

```
          bit numbers              |1|1|1 |1|1|1
          0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 1
```

Refer to the MPE V INTRINSICS COMMANDS MANUAL (P/N 32033-90007) for the layout of the FOPTION specifications under the FOPEN command. For our file the FOPTION bits say the following about this file:

This is a standard file.
File equations are allowed.
NOCCTL.

Records are fixed length.
The file is identified by its own file name.
The file is binary.
The file is in the old permanent domain.

16. LOGICAL RECORD SIZE:

Octal Value: 176000 (in negative bytes) or Decimal -102

17. BLOCK SIZE:

Octal Value: 001000 (in sectors) or Decimal 512

18. EXTENT INFORMATION:

Octal Value: 002000

```
          bit numbers              |1|1|1 |1|1|1
          0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
bit map   0  0 0 0  0 1 0  0 0 0  0 0 0  0 0 0
```

```
                          Bit Numbers
                          0|1|2|3|4|5|6|7
                          ---------------
Sector offset to data     0|0|0|0|0|1|0|0        Decimal 4

                           Bit Numbers
                           1|1|1|1|1
                           1|2|3|4|5
                           ---------
Number of extents -1       0 0 0 0 0              1 extent
```

The sector offset to data shows where the data begins relative to the file label. The number of extents shows the number of extents allowed for this file, but it does not show the number allocated at this time.

19. LAST EXTENT SIZE:

Octal Value: 000044 (Sectors) or Decimal 36

20. EXTENT SIZE:

Octal Value: 000044 (Sectors) or Decimal 36

This is the extent size, in sectors, for all extents of this file with the exception of the last extent. This logical word may have a value from 1 to 65535. This will limit a file size to 2097120 sectors.

21. END OF DATA POINTER:

Octal Value: 000000 000010 or Decimal 8

This double word integer is a pointer which contains the record number of the next actual record past the end of file, in other words *record at end of File +1.*

22. WORD ADDRESS OF FIRST EXTENTS:

Octal Value: 001025 040144

```
                                  bit numbers              |1|1|1 |1|1|1
                                  0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
                                  ---------------------------------------
(001025)   Bit map for word 1 - 0 |0|0|0 |0|0|1 |0|0|0 |0|1|0 |1|0|1
(040144)   Bit map for word 2 - 0 |1|0|0| 0|0|0 |0|0|1 |1|0|0 |1|0|0
```

```
                          01234567
                          --------
Volume table index for    00000010          Decimal 2  Extent 1 resides on
  first extent                                           volume 2
```

```
                    |1|1|1|1|1|1|1 |W O R D 2          |1|1|1|1|1|1|1
                   8|9|0|1|2|3|4|5 |0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5
Address            ---------------------------------------------------
on Volume          0|0|0|1|0|1|0|1  |0|1|0|0|0|0|0|0|0|1|1|0|0|1|0|0
                   or (Octal) %5240144
```

The first extent resides on Volume 2 at octal address %5240144. Since there are no more extents, there are no more values in the label.

23. FILE RESTORE TIME:

Octal Value: 013404 026400

```
                                  bit numbers              |1|1|1 |1|1|1
                                  0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
                                  ---------------------------------------
(013404)   Bit map word 1 - 0  0 0 1  0 1 1  1 0 0  0 0 0  1 0 0
(026400)   Bit map word 2 - 0  0 1 0  1 1 0  1 0 0  0 0 0  0 0 0
```

```
                    01234567
                    --------
File restore hour   00010111        Decimal 23 or 11 pm

                          Bit Numbers
                              1|1|1|1|1|1
                          8|9|0|1|2|3|4|5

                          ---------------
File restore minute bit map 0|0|0|0|0|1|0|0 or Decimal 4
```

The file was restored at 11:04 pm.

These and all times are in CLOCK format discussed in the MPE INTRINSICS MANUAL (P/N 32033-90007) under the command CLOCK.

24. FILE RESTORE DAY:

Octal Value: 127157

```
          bit numbers                |1|1|1 |1|1|1
          0 |1|2|3 |4|5|6 |7|8|9 |0|1|2 |3|4|5
          ---------------------------------------
Bit map 1 |0|1|0 |1|1|1 |0|0|1 |1|0|1 |1|1|1

                        0|1|2|3|4|5|6|7

                        ---------------
File resore year        1|0|1|0|1|1|1|0       Decimal 87

                            1|1|1|1|1|1
                          8|9|0|1|2|3|4|5

                          ---------------
File restore day          0|1|1|0|1|1|1|1       Decimal 111
```

The file was restored on April 21, 1987.

These three words contain the time and date that this file was initially put on this system, whether or not it was by RESTORE or by building it. These and all dates are reported in Calendar Format described in the MPE INTRINSICS MANUAL (P/N 32033-90007) under the CALENDAR Command.

25. START OF FILE BLOCK NUMBER:

Octal Value: 000000 000000

These two words are valid only for variable length or message files; but contain the block number of the beginning of data for these file types.

26. BLOCK NUMBER OF LAST BLOCK:

Octal Value: 000000 000000

This is valid only for variable length or message files. It holds the number of current data blocks relative to the first block.

27. NUMBER OF OPEN/CLOSED RECS:

Octal Value: 000000 000000

This field is valid only for message files. It contains the number of open and close records in the file.

28. DEVICE CLASS NAME:

Octal Value: 042111 051503 020040 020040 DISC

This field is initialized when the file is created from the FOPEN, using the device specified. This field is accessed when more extents are allocated to ensure they are on the same device.

D. PROBLEMS THAT MIGHT RELATE TO FILE LABELS

File labels are *the* source of information on the characteristics of a file. Errors occur due to some unexpected circumstance. That unexpected circumstance is most often, after careful detective work, found to be misuse or error on the part of the user. They are seldom due to system or disc errors or to integrity problems.

Using the numbers assigned in TABLE A as a reference, the following is a cross index of errors that might be generated from each of the particular sections of the label and when they might occur.

1) GENERAL INFORMATION:

DIRECTORY POINTERS
NONEXISTENT ACCOUNT (FSERR 50)
NONEXISTENT GROUP (FSERR 51)
NONEXISTENT PERMANENT FILE (FSERR 52)
INVALID FILE REFERENCE (FSERR 54)

Normally, in these cases, either the file has not been correctly referenced or file label corruption could have occurred. Try :LISTF with the same file name to see if the file can be found with the expected characteristics.

OUT OF GROUP DISC SPACE (FSERR 61)
OUT OF ACCOUNT DISC SPACE (FSERR 62)

In attempting to keep a file on disc the user has exceeded the limits allowed for files in either his group or account. Use the :REPORT command to check to see if this might be so.

LOCKWORD VIOLATION (FSERR 92)

Either the user did not enter the lockword correctly or the lockword field is corrupted. A :LISTF,-1 or :LISTDIR LISTF;PASS will reveal the lockword.

DUPLICATE PERMANENT FILE NAME (FSERR 100)

A permanent file already exists with this name. Use a :LISTF to determine if this is true.

PERMANENT FILE DIRECTORY OVERFLOW (FSERR 103)

The directory structure cannot insert another entry at this area. Often this points to a naming convention that uses the same first letters at the beginning of the file name. To remedy this situation, a temporary solution would be to store off files in the offending account, purge them, or restore them to a different account. A more permanent solution is to change naming conventions and/or alter the directory size through a reload.

2) FILE SECURITY MATRIX:

When trying to run a program:

ILLEGAL CAPABILITY (FSERR 2) or (LOAD ERR 39)
together with
UNABLE TO LOAD PROGRAM TO BE RUN (CIERR 625)

LISTDIR is of great use if the above errors have occurred. Within LISTDIR, a LISTF will show the capabilities that a program has. Using this you can determine whether or not a program requires a capability that a user does not have (for example, privileged mode.)

OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 40)

LISTDIR, LISTSEC will show the file security matrix for the particular file in question. Does this user have this access to this file? Is it set up for the type of access that is being attempted?

When attempting to access a file:

PRIVILEGED FILE VIOLATION (FSERR 45)
SECURITY VIOLATION (FSERR 93)
USER IS NOT CREATOR (FSERR 94)

Check with :LISTF,2 to see if it is being accessed. Use LISTDIR LISTF;PASS to check to see who is creator and file code.

When trying to keep a file:

USER LACKS SAVE FILES(SF) CAPABILITY (FSERR 111)

The user does not have the ability to save files. Look to see what capabilities the user has with LISTDIR LISTUSER.

3) NATIVE LANGUAGE AND STORE/RESTORE BITS:

Error messages are received when trying to run a program:

OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 40)
EOF OR I/O ERROR ON PROGRAM FILE (LOAD ERR 63)
UNABLE TO LOAD PROGRAM TO BE RUN (CIERR 625)

These messages sometimes come from trying to access a file that has been locked by STORE (the store bits have been set by the store process and not successfully reset signaling completion). Then errors occur because the loader cannot gain write access to the program file due to the lock in the STORE/ RESTORE bits of the file label. If users want to run programs during store processes, then the program should be running or allocated before the STORE begins.

The program RESETBIT.PRV.TELESUP might be useful to restore usability of a file with the bits incorrectly set. This will be discussed in the next section.

4) FILE CREATION DATE:

5) FILE LAST ACCESS DATE:

6) FILE MODIFICATION DATE:

Inconsistency in these fields rarely gives an error as this is treated as an informational field. However, the :STORE command with DATE <= ACCESSDATE or >= MODIFYDATE might not store the files required if this field is corrupted. LISTDIR LISTF will allow these fields to be displayed so that they might be checked for accuracy.

7) FILE CODE:

In attempting to access a file:

PRIVILEGED FILE VIOLATION (FSERR 45)
SECURITY VIOLATION (FSERR 93)

Check to see if the file that a user is attempting to use is actually a privileged file (usually a data base).

8) FILE LABEL PRIVATE VOLUME INFORMATION:

In attempting to access a file:

DEVICE UNAVAILABLE (FSERR 55)
USER LACKS PRIVATE VOLUMES (UV) CAPABILITY (FSERR 112)
VOLUME SET NOT MOUNTED - MOUNT PROBLEM (FSERR 113)
VOLUME SET NOT DISMOUNTED - DISMOUNT PROBLEM (FSERR 114)
ATTEMPTED RENAME ACROSS VOLUME SETS - REJECTED (FSERR 115)

Check with :LISTF,2 or LISTDIR LISTF;MAP to see if this file resides on a Private Volume.

9) FLAGS:

This area of the file label is responsible for a number of problems to users if not interpreted correctly. Using :LISTF,2 or LISTDIR LISTF will reveal the settings in the file label.

A. BITS 0-1 Store/Restore bits (refer to #3 above)

B. BIT 2 Load Bit

C. BIT 3 Exclusive bit.

If any of these bits are set, then users trying to access the file will receive error messages such as:

EXCLUSIVE VIOLATION: FILE BEING ACCESSED (FSERR 90)
EXCLUSIVE VIOLATION: FILE ACCESSED EXCLUSIVELY (FSERR 91)
and a :LISTF,2 shows an * by the file name.

Sometimes an abort from a program was done leaving the file opened exclusively, or the program might be allocated, or a program that is currently loaded.

D. BITS 4/4 Device Subtype

E. BITS 8/6 Device type

If these bits do not contain a value (for a disc file %0, %1, or %3) an error message is output, such as:

INVALID OPERATION (FSERR 20)
OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 40)
OPERATION INCONSISTENT WITH DEVICE TYPE (FSERR 42)
INVALID DEVICE SPECIFICATION (FSERR 56)

F. BIT 14/1 Read bit

G. BIT 15/1 Write bit

The setting of these bits should not cause a problem even if they are in error. However, a :LISTF showing the file being accessed with no users might occur, see A, B, C above.

10. NUMBER OF USER LABELS:

This area of the file label might be a problem if the user is attempting to access a particular user label or if the number of user labels causes the first record, of the first block of data, to be misaligned according to the accessing program. The resultant error, in either case, would be garbage data retrieved.

11. MAXIMUM NUMBER OF LOGICAL RECORDS:

Since this is a capacity for this file, if this field is corrupted the user might not be able to write more records into a file or records added might be placed in an area not really within the boundaries of this file.

Possible errors in the first case:

END OF FILE (FSERR 0)
OUT OF DISC SPACE (FSERR 46)

In the second case, errors would be received on attempts to later access records in this file:

FREADDIR ERROR
LOST FREE SPACE IN (FILENAME)
CRITICAL READ ERROR ON (FILENAME)
CONDITION WORD 13 RETURNED BY DBGET (DIRECTED END OF FILE)

12. FILE CONTROL BLOCK VECTOR:

If the file is opened and this pointer is incorrect (unless there is a 0 here, which can be used to get around this as a problem) then access to the file by multiple users will be pointing to an incorrect address in memory and users will receive this error:

INVALID MEMORY ADDRESS (FSERR 37)

13. CHECKSUM:

If the checksum is calculated and determined to be incorrect then the following error message is received and access to the file is denied:

INVALID FILE LABEL (FSERR 108)

14. COLD LOAD IDENTITY:

If the coldload ID is in error it is ignored.

15. FOPTIONS SPECIFICATIONS:

If the FOPTIONS of the file label and the opening program do not match then the file label will override the FOPEN options used by the program. File type mismatches in the calling programs FOPTIONS might generate an error:

INVALID FILE TYPE SPECIFIED IN FOPTIONS (FSERR 9)

16. LOGICAL RECORD SIZE:

In attempting to write to a file, an inconsistency in the program record description and the file label record size disagree. When this is true then:

INVALID RECORD SIZE SPECIFICATION (FSERR 10)
or
WRITE EXCEEDS RECORD SIZE (FSERR 43)

17. BLOCK SIZE:

When attempting to write to a file if an inconsistency occurs then the errors look like:

INVALID RESULTANT BLOCK SIZE (FSERR 11)

18. SECTOR OFFSET TO DATA:

At times this can be a very useful pointer to an integrity problem on the system. If a problem exists then the error received might be:

ADDRESS OF FILE LABEL AND FIRST EXTENT DO NOT MATCH

For further information see #22 below.

19-20. (THESE SECTIONS HAVE BEEN INTENTIONALLY EXCLUDED)

21. END OF FILE POINTER:

Incorrect information in this field, which is the actual number of records in the file, might result in the same conditions as the errors discussed in #11 above.

22. BEGINNING ADDRESSES FOR EXTENTS:

In attempting to access a file an error is received, such as:

INVALID FILE LABEL (FSERR 108)
or
The same error received in #18 would occur.

ADDRESS OF FILE LABEL AND FIRST EXTENT DO NOT MATCH

Normally these indicate that there is an integrity problem with the file label. There are utilities present in the TELESUP account to help diagnose this problem. These utilities are discussed in the following section.

23-24. FILE STORE/RESTORE TIME AND DAY:

Normally these fields do not yield errors even if they might be corrupted since they are informational fields. It may be that the user will see dates in this area that are suspicious, however, often on further investigation you might find that they have been properly initialized by some process.

25. START OF FILE BLOCK NUMBER:

26. BLOCK NUMBER OF LAST BLOCK:

27. NUMBER OF OPEN CLOSED RECORDS:

Remember that these fields are valid for message or variable length files, so they would only be checked if these type files are in use. If this area is not correct, then data might be skipped over or pointed to incorrectly and give errors such as described in #11 above.

28. DEVICE CLASS NAME:

This field might present a problem if corrupted in the same manner as 1 above, since the file cannot be found by the operating system, but, more likely errors such as:

UNABLE TO LOAD PROGRAM TO BE RUN (CIERR 625)
or
OPERATION INCONSISTENT WITH ACCESS TYPE (FSERR 42)

E. DIAGNOSES OF FILE LABEL INTEGRITY

The following programs reside in the Telesup account. They are provided for use by Hewlett-Packard on your system. Hewlett-Packard is not responsible for its use or misuse.

Diagnoses of a bad file label is done most simply by running BADLABEL.PRV.TELESUP on a system with no one accessing any of the files. BADLABEL verifies what it can of file labels on the system which includes checking the disc free space maps and the extent maps of the files. To run this program the user must have Privileged Mode capability.

To run Badlabel:

:RUN BADLABEL.PRV.TELESUP(,NOTAPE)

If the NOTAPE option is *not* taken then cross checking to determine whether or not more than one file points to the same area of disc. This program will show which extents are in error from the maps in the label.

F. RECOVERY FROM BAD LABELS

DIRPUR.PRV.TELESUP will remove a file entry from the directory. To run this program the user must have privileged mode capability. It is important to remember that this program will not return disc space to the system so a COLDSTART, RECOVER LOST DISC SPACE must be done to do so.

To purge an entry from the directory with DIRPUR:

:RUN DIRPUR.PRV.TELESUP

DIRPUR will then prompt:

INPUT FILE NAME <<<<<<PUT IN THE FILE NAME>>>>>>>>>>
INPUT GROUP NAME <<<<<PUT IN THE GROUP NAME>>>>>>>>>
INPUT ACCOUNT NAME <<<<<<<<PUT IN THE ACCOUNT NAME>>>>>

To EXIT press (RETURN).

FLUTIL.PRV.TELESUP will also allow a user with Privileged Mode and System Manager capability to examine a file label and, if desired, rewrite it or purge it. To purge a file through FLUTIL identified as damaged:

:RUN FLUTIL.PRV.TELESUP

FLUTIL will then prompt:

?FILENAME= <<<<PUT IN THE FILE NAME HERE>>>>
?FUNCTION=DD (FOR DIRECPURGE)
?FUNCTION=E or X

RESETBIT.PRV.TELESUP will allow a user with Privileged Mode to reset the store bit in a file label on a single file or in whole groups and accounts. It will also allow the user to change the creator of a single file or groups and accounts. There are two entry points, one for resetting the store/restore bits, the other for changing the creator. When running this an "@" may be used to signify groupings. To reset the store/restore bits:

:RUN RESETBIT.PRV.TELESUP

Resetbit will then prompt:

FILE?<<<<<<<<<INSERT FILE NAME>>>>>>>>>>
GROUP?<<<<<<<<INSERT GROUP NAME>>>>>>>>>>
ACCOUNT?<<<<<<INSERT ACCOUNT NAME>>>>>>>>
FILE?<<<<<<<<<IF YOU WISH TO EXIT PRESS RETURN>>>>>>>>>>>>

To change the creator:

:RUN RESETBIT.PRV.TELESUP,CREATOR

Resetbit will then prompt:

NEW CREATOR NAME?<<<<<<ENTER THE NAME YOU WISH TO MAKE CREATOR>>>

Then the same scenario as above.

BACK ISSUE INFORMATION

Following is a list of the Application Notes published to date. If you would like to order single copies of back issues please use the *Reader Comment Sheet* attached and indicate the number(s) of the note(s) you need.

Note #	Published	Topic
1	2/21/85	*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	*Plotter Configuration Guide*
4	4/15/86	*Printer Configuration Guide - Revised*
5	5/01/86	*MPE System Logfile Record Formats*
6	5/15/86	*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	3/15/87	*Private Volumes*
25	4/01/87	*TurboIMAGE: Transaction Logging*
26	4/15/87	*HP 2680A, 2688A Error Trailers*
27	5/01/87	*HPTrend: An Installation and Problem Solving Guide*
28	5/15/87	*The Startup State Configurator*
29	6/01/87	*A Programmer's Guide to VPLUS/3000*
30	6/15/87	*Disc Cache*
31	7/01/87	*Calling the CREATEPROCESS Intrinsic*
32	7/15/87	*Configuring Terminal Buffers*
33	8/15/87	*Printer Configuration Guide*
34	9/01/87	*RIN Management (Using COBOLII Examples) (A)*
34	10/01/87	*Process Handling (Using COBOLII Examples) (B)*
35	10/15/87	*HPDESK IV (Script files, FSC, and Installation Considerations)*
34	11/01/87	*Extra Data Segments (Using COBOLII Examples) (C)*
36	12/01/87	*Tips for the DESK IV Administrators*
37	12/15/87	*AUTOINST: Trouble-free Updates*
38	1/01/88	*Store/Restore Errors*
39	1/15/88	*MRJE Emulates a HASP Workstation*
40	2/01/88	*HP 250 / 260 to HP 3000 Communications Guidelines*
41	4/01/88	*MPE File Label Revealed - Revised 6/15/88*

HEWLETT PACKARD RESPONSE CENTER QUESTIONS & ANSWERS

HP 3000 Questions Commonly Received by the North American Response Centers

Q. I would like to set my tape drive up in such a way that I can use it as auto-reply or require an operator reply, depending on the circumstances. How can I do this?

A. When you configure the tape drive, you will need to give it at least two device class names. One of the names should be unique to the system, such as AUTOTAPE. The other device class name should exist on the tape drive and at least one other device. For instance, add the device class REPTAPE to the tape drive and the job tape pseudo-device. Configure the tape drive for auto- reply, either by using subtype 8, 9, 10, or 11 on releases of MPE before T-Mit (G.01.00) or by answering "Y" or "YES" to the AUTO REPLY? prompt in SYSDUMP or INITIAL on releases after T-Mit.

If you use the device class names given above, then when you want to use the tape drive in auto-reply mode, you specify ;DEV=AUTOTAPE in the file equation for that tape drive. If you want to require an operator reply, specify ;DEV=REPTAPE. A reply request will be printed at the operator console asking which of the two logical devices with class name REPTAPE you wish to use.

*****

Q. Can I use the new 700/41 terminal on my HP3000?

A. The 700/41 terminal is not supported on the HP3000. The 700/41 is a multi-personality ASCII terminal, and will function on the HP3000 as an entry-level ASCII terminal would, executing MPE commands and running programs. However, the escape sequences which the 700/41 recognizes are different from those used by HP mode terminals like the 2392 or 700/92. This means the 700/41 can not support features such as block mode. For example, the escape sequence ESC &, which is used to start all display enhancement sequences on HP terminals, puts the 700/41 into its protected mode. Therefore, this member of the 700 series is not supported for use on the HP3000.

*****

Q. I would like to use a device class name twice in the I/O configuration for one of my devices. SYSDUMP and INITIAL, however, do not allow you to duplicate class names while adding a new device. How can I do this?

A. In order to add duplicate class names, use the "Class changes" section of the SYSDUMP/INITIAL dialogue. In this section, first delete the class, and then add it back in with the ldev numbers of all devices using that class name. List each ldev number as many times as it will be using that device class name.

For example, if you want to have the class "DISC" once on ldev 1 and 3, but three times on ldev 2, you should do the following in the SYSDUMP/INITIAL dialogue:

more...

```
ANY CHANGES?  Y
     <answer intervening questions with carriage return>
I/O CONFIGURATION CHANGES? Y
     <answer intervening questions with carriage return>
CLASS CHANGES? Y
LIST CLASSES?
     (answer "Y" if you wish to see a list of classes and related ldevs)
DELETE CLASSES? Y
CLASSES? DISC
ADD CLASSES? Y
CLASS NAME? DISC
LOGICAL DEVICE #'S? 1,2,2,2,3  (ALL ldevs using that class must be listed)
SERIAL DEVICE CLASS? N  ("Y" for cartridge tapes and other serial discs)
FOREIGN DISC CLASS? N
CLASS NAME?  (press carriage return and continue the SYSDUMP dialogue)
```

*****

Q. Can I spool my 7550 plotter on the 3000?

A. Starting with the Vdelta1 operating system, the 7550 is supported as a spooled device in a direct connect configuration from an ATP/ADCC port. Modems, multiplexors or other third party devices are not supported.

On the HP 3000 the ATP software version must be G.51.02 or later and the ADCC version must be G.51.31 or later. The plotter must be configured as type 32, subtype 14, termtype TT7550, record width 66, speed 960 (required), mode S only, driver HIOASLPL0 for the ATP and HIOASLPL2 for the ADCC, device class S7550. In addition, Pass 3 or greater firmware is required for ATP ports. This can be verified by running TERMDSM and typing display, entering ATP ldev and HWDIT for hardware dit. If the value of HW'PCC'DATE is 0 or 1 then you will need to contact your CE for a firmware upgrade.
On the 7550 plotter make sure it is set up in stand-alone configuration with the following front panel settings: Monitor Mode OFF, Data Flow REMOTE STANDALONE, Bypass mode OFF, Handshake XON/XOFF, Connection DIRECT, Duplex FULL, Parity 8-BIT OFF and Baud rate 9600.

You also need to issue the command :HEADOFF ldev to suppress the spooler banner page.

*****

Q. I tried to cold load my system received the error message "SYSTEM HALT 6" or "CHANNEL PROGRAM TIMEOUT". What is this and what should I do?

A. On a system LOAD (COLDLOAD, RELOAD or UPDATE), the operating system is loaded from magnetic tape media on the tape drive. Therefore, it is critically important to make sure that the coldload tape is loaded and on-line before 'LOAD' is entered at the M> or -> prompt. If the tape drive is not already on-line when 'LOAD' is entered, this is one possible cause for SYSTEM HALT 6. SYSTEM HALT 6 is an indication the system is having problems trying to read the cold load tape. The tape drive being off-line is one common cause. Note that on MICRO, S/37, 6X,7X SYSTEM HALT 8 can indicate the same problem.

more...

Another possible cause is the DRT specified as the LOAD device is not the tape drive. On the S/6x and S/70 the LOAD device is displayed in the banner on the top of the console terminal. The number is in the format x,yy,zz where x = numbers from 0 - 2, y = numbers from 1 - 15, and z = numbers from 0 - 7. The LOAD device DRT can be calculated with the following formula: DRT= (x*128)+(8*yy)+zz. This value should match the DRT number for the tape drive in the latest SYSDUMP I/O configuration. If the LOAD device specified in the console banner does not match, use the above formula and determine what the values should be for x, y and z. Then type in 'LOAD:=x,yy,z' at the M> prompt.

On the S/37, MICRO/3000, MICRO/XE, the LOAD device can be determined by entering 'TEST' at the 'type H for HELP->' prompt (ctrl B). In the test mode, enter IO and it will display the LOAD channel and device. The LOAD DRT can be calculated with the following formula: DRT=8*channel number+ device number. If this calculated DRT number does not match the DRT number for the tape drive, as shown in a SYSDUMP I/O configuration, then use the 'LOAD channel #,device #,C' command to correct this situation.

For S/3x, S/4x, and S/5x, this is slightly different. The LOAD device is specified by two thumbwheel switches located on the front side of the CPU. The front of the CPU will have three sets of thumbwheel switches: START, DUMP, and LOAD. Each set of thumbwheel switches consist oftwo switches. The left one indicates channel number and the right one indicates device number. The DRT number is calculated using the following formula: DRT=8*channel number+device number. Check this calculated DRT with the latest SYSDUMP I/O configuration and make sure it is the same as the DRT for the tape drive. If it is not correct, use the above formula to determine the correct channel and device number, and set the LOAD thumbwheels accordingly. Another possibility is there maybe a hardware problem with the tape drive. To verify this, mount and put on-line the Diagnostic Utility Subsystem (DUS) tape on the tape drive. Enter 'LOAD' at the M> or -> prompt. If the DUS.tape boots up correctly, then the problem is not with the system accessing the tape drive, but most likely the problem is with the cold load tape. Attempt to COLDLOAD the system from a cold load tape known to be good. If DUS does not boot up, power off the tape drive and the CPU. The hardware may be in an unknown state and power cycling the hardware may clear this. Wait 30 seconds and turn the power back on to the CPU and the tape drive. Now, try to LOAD the DUS tape again. If this works, try to cold load system again. If this last effort does not work, there is good possibility there is a hardware problem. Before calling hardware support, verify that the HPIB cables are not loose. Use IOMAP to verify the hardware recognizes that tape drive is connected to the system (enter the IOMAP command at the -> prompt of S/3x, S/4x, S/5x. For the S/37, MICRO/3000 and MICRO/XE type test at the -> prompt and then type IOMAP (enter EXIT to return to the prompt). For S6/x and S/70 DUS needs to be loaded to get IOMAP). If these possible resolutions have not corrected the situation you may require assistance from the response center.

* * * * *

Q. I need to backdate from V-Mit (G.03.00) to a previous release of MPE, and need to know how to undo the changes Security Monitor made. How can this be done?

A. To disable the security changes made by Security Monitor:

:RUN SECCONF.PUB.SYS,RESET

When prompted for an option, choose the HARD RESET.

more...

Q. I do not have a DUS (Diagnostic Utility Subsystem) tape. Is this tape important? How can I create a new DUS tape?

A. The DUS tape can be critical for recovering data or diagnosing problems in an emergency. Every site should have a current DUS tape. Series II/III/30 users receive DUS tapes with their V/R update package. Other users create a new DUS tape during the AUTOINST update process.

If you need to create a new DUS tape manually, you may do so as follows:

1) Check to make sure the group HP32231.SUPPORT exists on your system. If it has been purged, create the account SUPPORT with IA, BA, PM and ND capability, the group HP32231.SUPPORT with IA, BA, PM capability, and the user FIELD.SUPPORT with ND capability, and restore @.HP32231.SUPPORT from your latest FOS tape.

```
EXAMPLE:
   :HELLO MANAGER.SYS
   :NEWACCT SUPPORT,FIELD;CAP=IA,BA,PM,ND
   :ALTUSER FIELD.SUPPORT;CAP=IA,BA,ND,AM
   :NEWGROUP HP32231.SUPPORT;CAP=IA,BA,PM
   :FILE CT;DEV=CTAPE        (note: use the class name or ldev number
                                of your tape drive )
   :RESTORE *CT;@.HP32231.SUPPORT
      (mount the FOS tape, put it on line, and reply to the tape
       request if necessary)
```

2) Sign on the system as FIELD.SUPPORT in the group HP32231.

3) Run the program COPYDUS to create the DUS tape.

4) COPYDUS will prompt you for the type of tape (CARTRIDGE TAPE, MAG TAPE, or FLOPPY DISC). Type in the correct name, then mount a tape in the tape drive with a write ring/write enabled. Reply to the tape request if applicable.

```
Example:
    :HELLO FIELD.SUPPORT,HP32231

    :RUN COPYDUS

    DUS COPY ROUTINE REVISION 1.06
    ENTER MEDIA TYPE(FLOPPY DISC, CARTRIDGE TAPE, MAG TAPE): CARTRIDGE TAPE

    INSTALL SCRATCH TAPE
    Reply to 'CART' on the system console.
           (reply to tape request if applicable)
    BEGIN TRANSFER OF DATA
    OPENED FILE: TAPEDUS.HP32231.SUPPORT
    BEGIN VERIFICATION OF DATA

    END OF PROGRAM
```

more...

5. Once the program has terminated successfully, remove the tape from the tape drive, label it, and keep it.

It is not necessary to have a different DUS tape for each system. However, you should make a new DUS tape every time you update MPE, since newly supported devices (such as the 7937/8 and 7957/8 disc drives) will only be supported on newer versions of DUS. The special DUS tapes which come with the series III/30 should be kept separate from other DUS tapes, since these systems use special DUS software.

* * * * *

Q. When do you take a string dump (Series 6X and 70 only) and a memory dump?

A. System Failures:

A memory dump should always be taken for a System Failure. System Failures are listed in section "MPE Message System" of the System Operation and Resource Management Reference Manual (SORM).

A string dump should be taken for all System Interrupt errors, excluding system failures, followed by a memory dump. For example, System Halts, Hangs, MPL (Micro code Program Load) errors, SYSSTOP errors, Halts. If a memory dump is taken first, the string dump will be invalid since the system registers will be reset. Hardware errors are listed in section "System Controls and Special Functions" of the SORM.

In summary, always take a memory dump, and under the conditions listed above, take a string dump. If a string dump is needed, remember to take the string dump *before* doing the memory dump.

* * * * *

Q. Whenever I use a new tape on my tape drive, I receive the error message "LDEV n I/O ERROR IGNORED DURING AVR. I/O STATUS %14". What is this message? Does this mean that my new tapes are bad?

A. The acronym "AVR" stands for Automatic Volume Recognition. During AVR, the tape drive reads part of the tape to determine it's density and get some information about it's tape label (if a label exists). If it can not determine the density of the tape, the above message will be issued.

This AVR message may appear when a tape is first mounted under the following circumstances:

- a brand new tape (the tape has no previous information written on it)

- a tape that was written at a density that cannot be read by the current tape drive (for example: trying to read a 6250 bpi tape on a 7979 tape drive)

- a problem with the tape or tape drive, such as dirty tape heads or a bad tape.

In most cases, if you are writing to a tape, the AVR message may be ignored. If you wish to read a tape and you receive the AVR message, you should clean the heads and check to make sure you are using the proper tape. If the problem persists, especially if you cannot read any tapes, you should contact your local hardware support team.

more...

Q. What escape sequence is used to set my Thinkjet to compressed print?

A. To set a Thinkjet printer to compressed print use: ESC &k2S. To set the printer to normal print use: ESC &k0S.

* * * * *

Q. During my update, I received an error INSTWARN #12 from the AUTOINST program, and I can not find it in any documentation. What does this error message mean?

A. INSTWARN #12 means the same thing as INSTERR #19, which is an error trying to locate the file SUPPACCT.PUB.SYS. The SUPPACCT file is used for creating the SUPPORT account as part of the AUTOINST process. To eliminate the error, restore the file SUPPACCT.PUB.SYS from the FOS tape and run AUTOINST again.

* * * * *

Q. My system line printer is configured as LOGICAL DEVICE 6, and DEVICE CLASS LP. What is the difference when I file equate FILE L; DEV=LP and FILE I; DEV=6 and perform the following:

```
FCOPY FROM=FILE1; TO=*L
FCOPY FROM=FILE1; TO=*I
```

A. Externally to the user, the two operations outlined above perform the same thing. They print FILE1 to the printer. However, within the MPE spooler, these are two different operations. By printing to *L(dev=lp) the document is sent to the DEVICE CLASS queue within the spooler. By printing to *I(dev=6) the document is sent to the LOGICAL DEVICE queue. Separate queues exist within the spooler for LOGICAL DEVICES and DEVICE CLASSES.

Why are there different queues that perform the same thing? The difference is how the MPE spooler handles multiple copies of printouts. Consider the following example:

FILE I; DEV=6,12,5 (Where 6 is L dev #, 12 is the outpri and 5 is the # copies)

```
FCOPY FROM =FILE1; TO=*I
FCOPY FROM =FILE2; TO=*I
FCOPY FROM =FILE3; TO=*I

FILE L;DEV=LP,12,5
FCOPY FROM=FILE1; TO=*L
FCOPY FROM=FILE2; TO=*L
FCOPY FROM=FILE3; TO=*L
```

Files printed to *I (dev=6) will appear in the following order: five copies of FILE1, followed by five copies of FILE2 and followed by five copies of FILE3. Five copies of each file are printed in sequential order.

more...

However, files printed to *L (dev=lp) will appear in the following order:

```
FILE1,FILE2,FILE3 : FILE1,FILE2,FILE3:
        o
        o
FILE1,FILE2,FILE3.
```

MPE spooler will collate multiple copies of output if they are printed to the DEVICE CLASS queue!

This has far reaching implications to your applications. Suppose you want to produce ten copies of ten different files to be distributed to your system branch managers. Instead of doing a lot of tearing and stapling, you can accomplish this by performing the following:

```
:OUTFENCE 14
:FILE L;DEV=LP, 13, 10
:FCOPY FROM=FILE1;TO=*L
:FCOPY FROM=FILE2;TO=*L
                o
                o
:FCOPY FROM=FILE10;TO=*L
:OUTFENCE 12
```

You now have ten sets of files in the order FILE1..FILE10 in collated order.

Note: Setting the OUTFENCE as described above is important for the proper collating of reports.

However, there are dangers which you should be aware of. Suppose while you are in the process of printing the above report, a user decides to print with an OUTPRI of 13. Another users file could be mixed somewhere with the above output.

There is not much you can do to avoid this except, to educate your users. Even if you set your files with OUTPRI 13, another users file could still appear with your output. One solution is to reserve OUTPRI 13 specifically for report collating. Educate your users to use 12 as the maximum OUTPRI.

* * * * *

Q. What is the difference between STARTSPOOL 6 and STARTSPOOL LP?

A. When a STARTSPOOL 6 is issued on the console, both the DEVICE CLASS QUEUE and the LOGICAL DEVICE QUEUE are spooled. However, when a STARTSPOOL LP is issued, only the DEVICE CLASS QUEUE is spooled. This means, LDEV 6 is presently running "HOT", if STARTSPOOL LP is performed. Your output will be spooled, if the file equated to the DEVICE CLASS (DEV=LP) and will run HOT if the file equated to the LOGICAL DEVICE NUMBER (dev=6). Your printer can then operate both as spooled and "HOT"! The secquence of commands to do this should be:

```
:OUTFENCE 14
:STOPSPOOL 6
:STARTSPOOL LP
:OUTFENCE 8
```

more...

Q. How can I lock my database so when I am doing a sysdump, it does not backup my data bases? I would rather use DBSTORE and DBRESTOR when it comes to backing up my database.

A. If you have ADAGER, you can use the LOCK function, otherwise you can: Run DBSTORE and specify your database name. When the console request appears, do not reply. Run your SYSDUMP or STORE program. When the second request appears on your console, reply to the second request. When the SYSDUMP operation encounters your database, it will not be able to store it because DBSTORE has locked the database. When SYSDUMP has completed, reply with 0 to the DBSTORE request to abort or go ahead and store your database now using the DBSTORE. If you have multiple databases, then run DBSTORE several times.

Place these commands in a stream file or UDC and you have an automated method for doing a backup and excluding your database files.

READER COMMENT SHEET

North American Response Centers
HP 3000 Application Note 41: MPE FILE LABEL REVEALED
(REPRINT** June 15, 1988)**

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

Is this Application Note technically accurate? ☐ Yes ☐ No

Are the concepts and wording easy to understand? ☐ Yes ☐ No

Is the format of this Application Note convenient in size, arrangement and readability? ☐ 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 Comments
Hewlett-Packard Worldwide Response Center
3300 Scott Boulevard
Santa Clara, CA 95054

FOLD FOLD