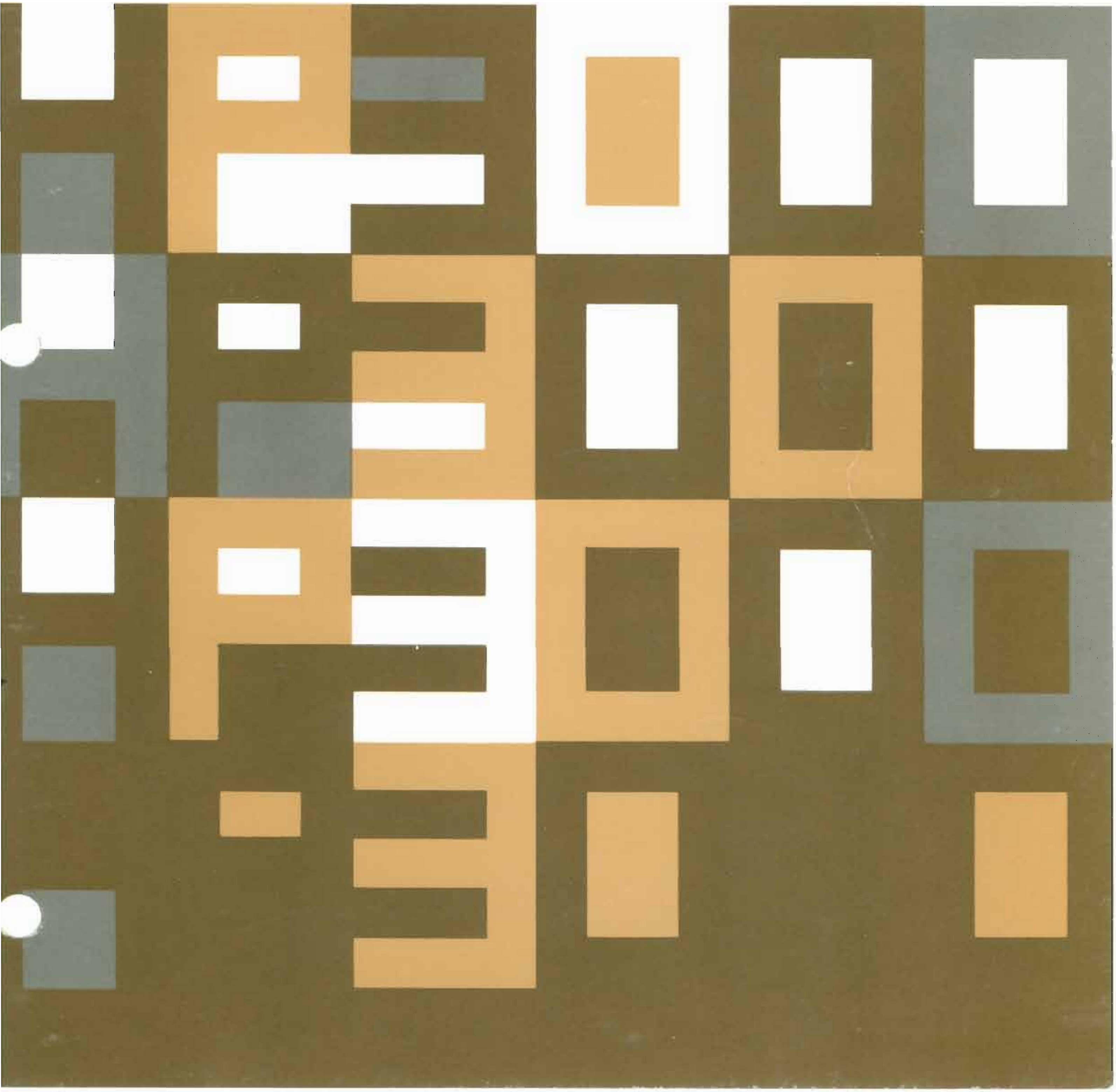
MPE IV C/D.01.00

HEWLETT PACKARD

ISSUE NUMBER 30

COMMUNICATOR



CONTENTS

SECTION I:
REGARDING THIS RELEASE

SECTION II:
GENERAL INFORMATION

Editor's Note

We at HP are always eager to know your thoughts concerning our products, and we make every effort to implement your suggestions to better tailor those products to your needs. This latest release of MPE (C/D.01.00), represents just such a step towards more useful, better quality software.

The COMMUNICATOR, too, has undergone some changes as a result of your suggestions. For those of you who maintain a library of the COMMUNICATOR, the issue date and MPE release number have been printed on the spine for easy referencing. Also in the interest of easy referencing, this issue has been organized into two topical sections. The first section contains only articles which are directly related to this HP 3000 software release. In this section you will find articles on the enlarged system directory and the enhanced STORE facility, as well as on the new :SET and =RECALL commands. Two new graphics packages being made available with this release, HPDRAW (which was used to create all of the illustrations in this issue) and HPEASYCHART, also have articles in this section, as do many other subsystems, utilities, and commands which are newly enhanced. There are many other important articles in this section, including an article that discusses several changes which, though small, are of great interest.

Two articles, OPT/3000 Enhancements and SEGMENTER Enhancements, would logically appear in this first section, but have been placed the very back of the COMMUNICATOR. These articles are printed on perforated pages so that they can be easily removed and inserted into your existing manuals as temporary supplements.

The second section contains articles which are not directly related to this release of HP 3000 software, but which will provide much useful information (which in many cases, you have requested in your reader comment sheets). Among the articles that you will find in this section is one detailing the use of labeled tapes, and one discussing file codes. This section also contains the Software Update listing, and the Catalog of Customer Publications.

We hope that you enjoy this issue, which was produced entirely with HP products, and that you find this new organization helpful. As in the last issue, you will find the Reader Comment Sheet on the very last page. We hope that you will continue to send us your comments on the COMMUNICATOR, so that we can continue to implement as many of your suggestions as possible.

Editor
HP3000 COMMUNICATOR
Computer Systems Division
19447 Pruneridge Avenue
Cupertino, California 95014

HEWLETT
PACKARD

SECTION I:
REGARDING THIS RELEASE

Introducing HP's Latest MPE IV Quality Release

by Sam Quezada, Computer Systems Division

As part of Hewlett-Packard's long standing commitment to quality, the Computer Systems Division (CSY) has taken a step to improve the quality of MPE. With this software release (C/D.01.00), the backlog of outstanding Service Requests (SRs) and Known Problem Reports (KPRs) has been substantially reduced. Additionally, two major enhancements have been included: the STORE facility within MPE has been enhanced in reliability and functionality, and the MPE system directory has been enlarged to greatly increase the number of files that can reside on the system. Also included with this software are several new products, and enhancements to existing products. These MPE changes, product enhancements and new products represent another step in our continuing efforts to imporve the quality of our product offering.

Quality Improvements

During the last five months the entire MPE team at CSY has temporarily suspended development on most enhancements to MPE. Instead the lab focused on quality. This MIT release (2244, which includeds *v.uu.ff* C.01.00 and D.01.00) represents over 70 engineer-months invested in quality. With this software, you will not see many major changes in function - but you will see many smaller changes which when added together will greatly increase the quality of MPE.

Software concerns arrive at Hewlett-Packard as Service Requests (SRs). These are classified as:

o KPRs (Known Problem Reports) - problems which will require correction.

o Requests for a change in functionality (Enhancement Requests).

o Duplicates of previous SRs or KPRs.

Over 2000 of your SRs have been resolved with this software release. Of these, over 460 resulted in functionality and reliability enhancements to MPE.

MPE Enhancements

Several MPE commands have been added or enhanced with this MPE release:

o The SETJCW command has been enhanced to support arithmetic manipulations for more sophisticated job control.

o The VINIT utility has been enhanced to support disc-to-disc backup fully.

- The new =RECALL command gives the console operator added control over operator requests.

- The new :SET command allows users to economize on paper and system resources by allowing them to delete job listings when not needed.

There are two major additional enhancements in this software release of MPE:

- Enhanced STORE facility - Several reliability enhancements have been incorporated. In the past, in the unlikely event that a tape failure was to occur within a STORE command, the entire STORE would have to be re-done. Should a failure occur in the new STORE, only the failing tape volume will have to be re-done. Furthermore, the code itself was enhanced to minimize the possibility of system failures.

- Enlarged system directory - There is a significant increase in the size of the system directory. System managers may configure up to 65K sectors for the system directory. This may provide up to a ten-fold increase in the number of files which can reside on the system.

New or Enhanced Subsystems

Within the product offering for the office, DSG/3000 (Decision Support Graphics/3000) has several new product features, and has been enhanced not only to support additional peripherals, but to provide support for two new, easy to use graphics packages, HPDRAW and HPEASYCHART, that are being made availble with this software release. TDP/3000 (Text and Document Processor/3000) has also been enhanced so that files from HPEASYCHART, HPDRAW, DSG/3000 and HPMAIL may be merged with TDP's text files.

Several system and programmer productivity products will be enhanced or are being made available for the first time. The VPLUS/3000 facility has been enhanced to take advantage of the forms cache capabilities of the 2624B terminal. OPT/3000 has been enhanced by incorporating several features requested in a recent users survey. We are also introducing HPToolset/3000, a COBOL-II programming development utility.

Detailed articles on all of these features may be found in this issue of the Communicator.

By providing a solid foundation of high quality software, this version of MPE will make it easier to develop and maintain future enhancements. As a result, we will be able to carry this quality through to future releases.

:STORE Command Enhanced

by Stan Sieler and Lou Bershad, Computer Systems Division

The MPE :STORE command has been enhanced on the C/D.01.00 release of MPE in an ongoing effort to better meet our customers' reliability and functionality requirements. STORE (and consequently SYSDUMP) now provides recovery from tape errors. Fileset handling capabilities have been extended, and the user interface has been made more "friendly". This article details the enhancements to STORE, and describes the effects on RESTORE.

Syntax

The following is an excerpt from the C/D.01.00 update to the System Manager/System Supervisor Reference Manual (30000-90014). For full details, please refer to the manual. Note that only the new parameters are defined in any detail here.

```
:STORE { [fileset [-fileset]] [, ...] } ;storefile
       { [indirectfile]               }

        [;SHOW=[showoption[,showoption[,...]]]]
        [;FILES=maxfiles]
                {<=accdate}
        [;DATE  {>=moddate} ]
                    {QUIT}
        [;ONERR[OR]={REDO}]
```

where

```
                [SHORT   ]
                [LONG    ]
  showoption is [DATES   ]
                [SECURITY]
                [OFFLINE ]
```

Expanded SHOW Option

The information reported by SHOW= has been expanded by the addition of the following options:

SHORT (Default.) For each file STOREd, the following information is printed on the user's terminal (default): file name, group name, account name, logical device number, reel numbers, file size (in sectors) and file code.

LONG — Displays same information as SHORT, plus: record size, filetype, EOF, file record limit, blocking factor, extents allocated and maximum extents of files STOREd. (If SHOW's list device has a record size greater than 100 characters, this option is selected automatically.)

DATES — Causes SHOW to display the creation date, last access date, and last modification date of files STOREd.

SECURITY — Causes SHOW to display the creator and file access matrix of all files STOREd.

OFFLINE — Causes SHOW output to be sent to the system lineprinter in addition to being displayed on the user's terminal. Note that output can be redirected to another output device by using a :FILE command. (Formal designator is 'OFFLINE'.) For example:

```
:FILE OFFLINE;DEV=PP;CCTL
```

Tape Error Recovery

In previous versions of STORE, any error encountered while writing to tape would cause the STORE to abort. Thus, if STORE encountered a tape parity error on the last file of the last reel of a large sysdump, for example, the entire STORE operation would have to be repeated, resulting in a loss of time and resources.

With the enhanced version, you can specify whether or not error recovery is desired by the use of the ONERROR keyword:

```
          {QUIT}
;ONERROR= {REDO}
```

If ONERROR is not specified, the defaults are REDO for unlabeled tapes, and QUIT for labeled tapes. If the QUIT option is selected, the STORE operation aborts upon sensing a tape error. The REDO option causes STORE to rewind the current tape reel, mark it as bad, request a new reel, and begin writing to the new reel at the point in the *fileset* list where the previous (bad) reel had started. This new tape reel completely replaces the bad reel. A message is sent to the operator's console advising that the just-rewound reel contains a tape error. ONNERROR=REDO cannot be specified for labeled tapes.

Improved Disc/Directory Error Tolerance

If a disc or directory error is encountered while building the list of files to be STOREd or reading a disc file for storing, the error is reported on the user's $STDLIST device along with the disc sector address, and the STORE continues. The file containing the error is prematurely closed and is not accessible to RESTORE. (RESTORE has been enhanced to continue after encountering such a file on a STORE tape.)

"Timely” Messages

The message-handling facility of STORE has been enhanced to print messages as they are generated, rather than displaying a list of messages at the end of the STORE operation. Interactive users can now monitor the progress of STORE and see error messages as they occur. Thus, you can abort STORE if, for example, you discover that an important file is "busy" at the time of the STORE. A message is displayed for each file which is successfully STOREd.

Improved Error Messages

Previously, when STORE encountered a problem of any kind, it aborted with the error message "ERROR WHILE WRITING TO TAPE - STORE STOPPED (CIERR 1000). Since this message could be generated by several different causes, diagnosis of and recovery from the problem was often difficult. The enhanced version of STORE generates an error message appropriate to the particular error encountered, making it easier to determine the correct recovery procedure.

A complete list of STORE error messages may be found in the November 1982 update to the Console Operator's Guide (32004-90002).

Indirect Fileset Specification

STORE now allows the use of indirect files to specify filesets and options. This feature is useful, for example, in the case where it is necessary to STORE the same files over and over, or when the Command Interpreter limitation of 268-character commands is a problem. When using an indirect file, each line can contain both filesets and options, but only options can appear after the first semicolon (;) on each line. The following example illustrates the use of an indirect file:

```
      :STORE !INDSTORE;*TAPE
```

where the file INDSTORE could contain

```
      @.PUB.SYS,@.@.KSE;FILES=10000
      @.@.SUPPORT;SHOW=LONG,DATES
```

The -fileset Parameter

The *filesetname* parameter has been expanded to allow for elimination of a particular fileset or subfileset with the use of the new *-fileset* parameter. For example, the command

```
          :STORE @.@.@ -@.PUB.SYS;*TAPE
```

would STORE all files on the system <u>except</u> those in the PUB group of the SYS account.

Implementation

The enhanced version of STORE is a separate program (STORE.PUB.SYS), rather than a set of procedures with no global memory as previously. Thus, it is advisable to :ALLOCATE STORE.PUB.SYS in order to speed up the loading process and reduce the use of system resources.

CAUTION

> If you currently have a file in PUB.SYS named STORE, it will be necessary for you to assign a different name to that file.

As a result of the new structure, STORE can now be invoked via the :RUN command. You can use the "INFO=" parameter to specify filesets and options; the STORE command will execute with no further interaction. An example follows:

```
      :RUN STORE.PUB.SYS;INFO="STORE @.@.@;*TAPE;SHOW"
```

If no "INFO=" parameters are specified, the "-->" prompt will appear. Acceptable responses are: a complete STORE command, an MPE command preceded by a colon, or EXIT.

Notes on JCWs

- o When invoking STORE via the Command Interpreter, e.g., :STORE ABC;*T, the system defined JCW, CIERROR, will be set to the value 1090 if the STORE aborts for any reason. You can check this value after the STORE aborts by entering the command :SHOWJCW.

- o When invoking STORE via the :RUN command, the system defined JCW "CIERROR" will not be reset, i.e. its value will remain unchanged if the STORE aborts.

Examples

```
:STORE AA,BB;*T;SHOW   <<default, no parameters>>

FILENAME.GROUP    .ACCOUNT    LDN  ADDRESS REEL  SECTORS CODE

AA        .PUB    .TRAINING     3%00075176    1       201 AA
BB        .PUB    .TRAINING     3%01007433    1        60 BB


FILES STORED:        2
FILES NOT STORED:    0
```

```
:STORE AA,BB;*T;SHOW=LONG

FILENAME.GROUP    .ACCOUNT   LDN  ADDRESS REEL  SECTORS CODE
           SIZE TYP      EOF     LIMIT R/B #X/MX

AA       .PUB     .TRAINING     3%00075176    1      201 AA
           128W FB       177         59   1  1/01
BB       .PUB     .TRAINING     3%01007433    1       60 BB
           128W FB        59         59   1  1/01

FILES STORED:        2
FILES NOT STORED:    0
```

```
:STORE AA,BB;*T;SHOW=LONG,SECURITY,DATES

FILENAME.GROUP    .ACCOUNT   LDN  ADDRESS REEL  SECTORS CODE
            SIZE  TYP     EOF    LIMIT R/B #X/MX  CREATED ACCESSED MODIFIED
AA       .PUB     .TRAINING     3%00075176    1      201 AA
           128W FB       177        200   1  1/01  4/06/82 10/06/82  4/06/82
           USER1    (R:ANY; A:ANY; W:ANY; L:ANY; X:ANY)
BB       .PUB     .TRAINING     3%01007433    1       60 BB
           128  FB        59          59  1  1/01 10/06/82 10/11/82 10/11/82
           USER1    (R:ANY; A:ANY; W:ANY; L:ANY; X:ANY)

FILES STORED:        2
FILES NOT STORED:    0
```

Expansion of Directory Size

by Sam Yamakoshi and Adrienne Frost, Computer Systems Division

Project Overview

The MPE directory's maximum size has been increased to 65,000 SECTORS, compared to the previous 6,000 sector maximum. This maximum directory size is applicable to both system and private volume discs on the HP 7920, HP 7925, and HP 7933.

Compatibility with Previous MPE Releases

The MPE changes which were made to support this new maximum size were designed with the intent of maintaining compatibility with prior MPE releases. The only restriction is that once the directory on a private volume or system disc has been increased beyond 6,000 sectors, only the C/D.01.00 MPE version can access that disc. Existing private volumes do not have to be reformatted to be accessed by the C/D.01.00 release of MPE.

Increasing the Size of the System Directory

A significant external change in the procedure to alter the directory size beyond 6,000 sectors deals with SYSDUMP. When you increase the size of the system directory beyond 6,000 sectors, SYSDUMP creates a temporary disc file 'SYSTDIRC' to contain a working copy of the new directory. This will also occur when decreasing the directory after it has been previously increased beyond 6,000 sectors. The size required for this temporary disc file is the number of sectors in the smaller of the new and old directories, and must be *contiguous* space. For example, increasing from 6,000 to 10,000 sectors requires 6,000 sectors of contiguous free space, as well as decreasing the directory from 10,000 to 6,000 sectors. If space is not available on any of the system domain discs for this temporary file, SYSDUMP will abort. Note that altering the directory from 384 sectors to 6,000 does not require this temporary disc space.

Copying the directory to a temporary file is necessary to convert the index and entry pointers (which are relative to the beginning of the directory) to reflect their new locations within the directory space. In the previous MPE directory, the bitmap (the free space table for the directory) was always 3 sectors in size in order to handle up to 6,000 sectors. This made the beginning (SYSACCTINDEX) of the directory indices and entries always equal to three, relative to the beginning. With a 65,000 sector maximum size, the directory requires from 3 to 33 sectors for this bitmap. SYSDUMP converts all directory pointers and copies the new directory along with the system to the tape or serial device. The approximate time required for this conversion is 4-6 accounts per minute. One test of 400 accounts required 75 minutes on an HP7920 disc.

The procedure for altering the size of the system directory beyond 6,000 sectors is as follows:

1. UPDate with MPE version C/D.01.00.

2. SYSDUMP *NULL, in order to find out the current directory size.

3. Run FREE2.PUB.SYS to ensure that there is at least the amount of available contiguous free space, as found in step 2, for the conversion.

4. SYSDUMP, altering the directory to the desired size.

5. RELoad the system disc with the cold load media created by SYSDUMP in step 4.

Following is an example of the SYSDUMP dialog to alter the system directory size to 60,000 sectors:

```
:FILE T;DEV=TAPE
:SYSDUMP *T

ANY CHANGES? Y
      .
      .
      .
DISC ALLOCATION CHANGES? Y
DIRECTORY USED = 109, MIN = 521, MAX = 1536.? 60000
      .
      .
      .
ENTER DUMP DATE?
```

Private Volume Directory

The directory of a new or scratched private volume master can be made up to 65,000 sectors in size via the INIT command in VINIT. There are no restrictions as to the size of a private volume directory in relationship to the system directory, i.e., it can be larger or smaller. A private volume directory which has been increased beyond 6,000 sectors cannot be accessed by a system which has a release of MPE that does not support expanded directories (any release of MPE prior to C/D.01.00). Also, once a private volume's directory size is has been established and the private volume is in use, the directory size cannot be altered without destroying the previous contents of the private volume due to the characteristic of the INIT command in VINIT.

Following is an example of the VINIT dialog to configure a private volume directory size to 60,000 sectors:

```
:DOWN ldev

VINIT C.01.00 (C) HEWLETT-PACKARD CO., 1978
>DSTAT ALL

    (The output will print here)

>SCRATCH ldev
>FORMAT ldev
>INIT PVVOL1,2,PVVOLL.PUB.SYS
 ENTER DIRECTORY SIZE (SECTORS - 384 TO 65000): 60000
>EXIT

END OF SUBSYSTEM
:UP 2
```

The New :SET Command Allows Deletion of $STDLIST

by Marie Weston and Jon Cohen, Computer Systems Division

Background

Every process has associated with it two files: $STDIN and $STDLIST. For most jobs, $STDIN is a spooled input file, and $STDLIST is a spooled output file. A spooled output file is a disc file that appears to a process as a printing device. Spooled disc files are then sent to the device when the device becomes available.

In many circumstances, the output of a successful job is of no interest. For example, suppose a job is streamed that will update files. If the job completes successfully, chances are you won't need the job's $STDLIST and if you can't get into, the MPE utility, SPOOK fast enough to delete the spoolfile, time and paper are wasted.

With the new :SET command, a job can now mark its $STDLIST for deletion when the job terminates.

How and When to Use The SET Command

To avoid receiving unwanted job listings, simply insert this command into the job stream:

```
:SET STDLIST=DELETE
```

This command can be placed anywhere between the JOB and EOJ statements since it simply sets a flag that marks $STDLIST for deletion upon job termination. It is more practical to place it at the end of the job stream; if the job fails, for some reason, the command will not execute and $STDLIST will print. You can then analyze your listing to see what went wrong.

You can also reverse the effect of the previous :SET command by entering into the job stream:

```
:SET STDLIST=SAVE
```

By selective use of these commands in a job stream, you can always be assured of getting a job listing when you need one and not wasting valuable time and system resources when you don't.

:SETJCW Command Enhancement

by Sara Mears, Computer Systems Division

The SETJCW command has been enhanced to allow simple addition and subtraction when assigning a value to a JCW. SETJCW will evaluate a simple expression and assign the resulting value to a JCW. This enhancement takes effect with the C/D.01.00 version of MPE, and applies to any HP 3000 running that software.

Syntax

The new syntax for the SETJCW command is

```
                             {+}
SETJCW jcwname char value [       value  ]
                             {-}
```

where the variables mean:

jcwname The name of the Job Control Word.

char Any special non-blank character except "%" or "-". Note that whatever character is chosen, it will be treated as "=", an assignment character.

value An octal or decimal number, a system defined value (e.g. FATAL, OK3) or the name of an existing JCW. Numbers are assumed to be decimal values unless preceded by a percent sign ("%"), which indicates an octal value.

All specified and calculated *values* must be in the range of 0 to 65535, inclusive. Internally JCWs are treated as logical values, not integers. Negative JCW values will cause the following error message to be displayed:

```
MAXIMUM JCW VALUE IS 65535 (CIERR 1712)
```

Examples of New Syntax of SETJCW

```
:SETJCW XX=5+6
:SETJCW YY=XX+1
:SETJCW ZZ=YY-XX
:SETJCW NN=NN+4
```

Potential Use

The enhanced :SETJCW command will permit more sophisticated job control than was previously available. For example, suppose you have a stream file which streams sixteen processes. Each time a process completes, it can set a bit in the same JCW. The first process may contain the command

SETJCW TEST=TEST+1 The second may contain the command

```
     SETJCW TEST=TEST+2
```

and so on until the sixteenth process contains the command

```
     SETJCW TEST=TEST+32768
```

When the job stream has completed, you can determine which processes completed successfully simply by inspecting the JCW. For each successful process, the corresponding bit in the JCW will be set.

New JCW Naming Restrictions

With this release, some modifications to JCWs were needed to clarify a potentially confusing situation. Both the :SETJCW command and the PUTJCW intrinsic now have the following restriction: JCW names can no longer be the same as any predefined JCW values. The system has predefined the following mnemonics as JCW values:

Value Name	JCW value
OK	0
WARN	16384
FATAL	32768
SYSTEM	49152

Further, these JCW mnemonics can be concatenated with a number to mean the addition of the mnemonic value and the number. For example, OK200 is a value of 200, and WARN32 is a value of 32800. Confusion can arise when a JCW is named with a string that is an acceptable predefined JCW value. Consider the following sequence:

```
     :SETJCW OK200 = 1982
     :SETJCW XYZ   = OK200
```

It is not clear whether the JCW "XYZ" should receive the value 200 or the value 1982. For this reason, an acceptable JCW name cannot also be an acceptable JCW predefined value. In the C/D.01.00, the command

```
     :SETJCW OK200 = 1982
```

would result with the following error message:

```
     JCWNAME CANNOT BE A VALID JCW VALUE (CIERR 1725)
```

UDCs Explained (and Extended)

by Jon Cohen, Computer Systems Division

User Defined Commands (UDCs) are useful and popular tools, but the implications of UDC internals are not widely understood. This article is intended to explain how UDCs are initialized and how they are invoked. This article also goes on to explain a new extension to UDC functionality being introduced in the C/D.01.00 release of MPE.

UDC INVOCATION

Each user has a UDC directory that contains all his UDC names as well as pointers into the files that define the UDCs. The creation of this directory will be discussed later. When a user is in the Command Interpreter (CI) and enters a command after the colon (:) prompt, the UDC directory is scanned, starting at the beginning of the directory, for the command string entered. If it is found, the UDC body for that command is read one line at a time, parameters are substituted into the line if appropriate, and the Command Interpreter is reentered at a special internal entry point. The UDC line is treated as a new command string, and is handled much the same as all other command strings: the UDC directory is scanned for a UDC that matches the command string, but this time, the UDC directory scan begins at the directory entry that follows the UDC currently being expanded. In this way, UDCs can call other UDCs as long as the called UDC was defined after the UDC being expanded.

If the CI scans to the end of the UDC directory without finding a match for the command string, the CI then tries to match the command string to an MPE command. This implementation allows some UDCs to call other UDCs, but does not allow any kind of UDC recursion.

This linear scan of the UDC directory is the heart of UDC invocation. This particular scan will allow UDCs with the same name to be executed at different times. Consider the following partial UDC file as an example:

```
*************************************
AUDC
COMMENT  This is udc AUDC.
*************************************
BUDC
AUDC
*************************************
                 .
                 .
                 .
*************************************
AUDC
COMMENT  AUDC called by BUDC.
*************************************
```

If the user enters "AUDC" as a command, it gets expanded and becomes the "COMMENT" line, and since there is no "COMMENT" UDC, the MPE command ":COMMENT" gets invoked with "This is udc AUDC." as its comment string. If the user enters "BUDC" as a command, it gets expanded to become the "AUDC" line. The rest of the UDC directory is then scanned, starting at the UDC defined after "BUDC"; the second "AUDC" is found and is expanded to become the "COMMENT" line, which in turn invokes the MPE ":COMMENT AUDC called by BUDC" command.

UDC INITIALIZATION AND UDC HIERARCHY:

UDCs are initialized when the user logs onto the system (and either user, account, or system level UDCs exist) or whenever the user executes a :SETCATALOG command. The system file, COMMAND.PUB.SYS, holds the UDC file names associated with each user; it also holds the UDC file names for account and system UDCs. At initialization time, all the UDC files needed by the user are opened, and a UDC directory is created for the user by scanning through all the UDC files.

As discussed earlier, the position of a UDC's entry in the UDC directory determines which UDCs can call it and which UDCs it can call. There is a UDC hierarchy imposed by the system: if there is more than one UDC file at any given level (user, account, system), UDCs are entered into the directory in the order they are found in the file. The files are opened and scanned in the order they appeared in the last :SETCATALOG command for that UDC level. All user UDCs are entered into the directory first, then all account UDCs are entered, and last, all system UDCs are entered.

It is not immediately apparent why user UDCs are entered first and system UDCs are entered last, but there is a definite reason. There are two major advantages in using UDCs: 1) they save lots of typing and may "hide" internal implementations, and 2) they are useful for enforcing site-specific security. The first advantage is obvious and needs no further discussion. The second advantage, that of site-specific security, is not so obvious.

A system (or account) manager might decide that a specific MPE command should be removed from the system (or account) users. If he wishes to remove, say, the :ABORTJOB command, all he need do is create a system (account) UDC called "ABORTJOB" that invokes a :COMMENT command. Thus, since the UDC directory is scanned before the MPE command is invoked, all users in the system (account) will encounter and execute the "ABORTJOB" UDC instead of the MPE command :ABORTJOB. A user cannot bypass this security feature by creating an "AJ" UDC that invokes ":ABORTJOB" since the user's UDC will actually invoke the "ABORTJOB" system (account) UDC in turn; the MPE :ABORTJOB command will never be executed.

If the UDC directory hierarchy was System-Account-User instead of User-Account-System, the user "AJ" UDC would bypass the system "ABORTJOB" UDC, and the MPE :ABORTJOB command would be executed instead. The present User-Account-System hierarchy allows system (account) managers to have the last word in creating their own security measures.

UDCs EXTENDED

In order to improve system security at logon time, logon UDC functionality has been extended in the C/D.01.00 release: at most one logon UDC at each level (User, Account, and System) will execute. Further, a system logon UDC will execute first, then an account logon UDC, and finally a user logon UDC. With this extension, system and account managers can use UDCs to enforce a site-specific security at logon.

Back by Popular Demand . . . =RECALL

by Ken Jordan, Computer Systems Division

With the C/D.01.00 release of MPE, we will be seeing the return of a command to the console interface that has been one of the most requested. In response to the many Service Requests (SRs), =RECALL will finally be returning to the console interface to complement the =REPLY command. It will work like the Command Interpreter's :RECALL, but can be executed in the same circumstances as any other A^c (CNTL A) command.

The Reply Information Table, which holds all information about processes which need an operator REPLY, has also been changed. The maximum number of entries has been increased from 22 to 39. A queue has been added for processes that try to enter the table when it is full. This will allow 39 processes to be awaiting a REPLY and another 52 processes to be queued awaiting entry into the table. When the table and queue are both full, any attempt to enter the table will be rejected.

When a user is queued because the Reply Information Table is full, the message:

```
     THE REPLY INFORMATION TABLE IS FULL, REQUEST IS QUEUED
```

will be displayed on the user's terminal, and the message:

```
     10:05/#S2/32/THE REPLY INFORMATION TABLE IS FULL, 1 REQUEST(S) QUEUED
```

will be displayed on the console. When the process is awakened the message:

```
     THE REQUEST FOR AN OPERATOR REPLY IS NOW PENDING
```

will be displayed on the user's terminal and the usual request message will come up on the console. If the table and queue are both full, the user will receive the message:

```
     THE SYSTEM TABLES FOR A REPLY ARE FULL, ATTEMPT REJECTED
```

and the operating system will remove the entry. The console will be sent the message:

```
     10:10/#J58/54/THE REPLY INFORMATION TABLE AND ASSOCIATED QUEUE ARE BOTH
     FULL
```

A =SHUTDOWN will work, even if there are queued entries, by aborting each process so that it will log off. :ABORTJOB and =ABORTJOB also will work on any queued processes.

Identifying Systems with MPE's System Message Catalog

by Jon Cohen, Computer Systems Division

We in the Operating Systems R&D Lab have our computer workload distributed between several HP 3000s connected together via DS/3000. Typically, an engineer has several sessions active on several different machines. If a message should arrive via the :TELL command, the recipient would be unable to determine the source unless the sender explicitly identified the issuing machine.

We have implemented a simple solution to this problem: we placed a system identifier into selected messages in the system message catalog. This is the method we used:

1. Log on as MANAGER.SYS.
2. Enter EDIT/3000 or TDP/3000 text editor.
3. Text in CATALOG.PUB.SYS.
4. Modify selected messages (see below).
5. Keep the workfile in a new file, say "TEMPCAT".
6. Use a :FILE equation to equate "INPUT" to the new file
   (e.g. :FILE INPUT = TEMPCAT).
7. Run MAKECAT.PUB.SYS, BUILD to install the new catalog.

The messages we chose to contain the system ID are:

$SET Number	Message Number	New Message Text
1	9	(*system ID*) OPERATOR WARNING: !
7	3	FROM (*system ID*)/!/!

These are the messages used by the :WARN and :TELL commands. Thus, if we use "Blondie" as our system ID, a message from the operator may look like this:

```
FROM (Blondie)/#S1 OPERATOR.SYS/PLEASE LOG OFF
```

Of course, we use a different system ID for each of our timeshare systems. Since we tend to perform system updates on our machines often (we test our new software on our timeshare machines before we release it to our customers), the steps needed to modify each system's catalog are stored in a :STREAM file.

Small MPE Improvements Worth Noting

by MPE Team, Computer Systems Division

A number of miscellaneous changes have been included in the C/D.01.00 release of MPE. In addition to those detailed in other articles in this issue, the following changes warrant your attention.

:ALTJOB Command

The user version of the :ALTJOB command, which can be executed when the JOBSECURITY is set to LOW has been changed to allow only integers between 1 and 13 as valid INPRI values. The values 0 and 14 are no longer accepted unless the user has the AM capability.

:SHOWALLOW Command

Previously, unless the user logged onto the console had SM capability, the :SHOWALLOW command would only display commands ALLOWed to the *user.group.account* currently logged on. Thus, the operator had no way to examine the results of any :ALLOW commands issued from the console. The user at the console can now execute :SHOWALLOW for any and all users. The command is breakable, but cannot be RESUMEd after BREAK. A BREAK causes execution of the :SHOWALLOW to terminate.

Change to :FILE Command

Due to some confusion arising from the use of lockwords specified in a :FILE equation, the specification of lockwords is no longer permitted in :FILE formal designators. User expectations were that the lockword assigned to a formal designator would be transferred to the actual designator and that this was considered to be the name of a particular file equation. However, such names do not have lockwords associated with them. In MPE there is no need for lockword type security for file equations since these equations are only accessable within the job/session's process tree.

Formal designators can appear in one of two places in a :FILE command, e.g.:

```
     :FILE formaldes [;(other parms, e.g. DEV=)]
                     [= filename               ]
                     [= *formaldes             ]

     where formaldes = name[.group[.acct]], and
           filename  = name[/lockword][.group[.acct]]
```

The existence of a lockword in either *formaldes* position will now result in the following error message:

```
     NO LOCKWORDS ALLOWED IN :FILE FORMAL DESIGNATORS.  (CIERR 199)
```

Note that, since lockwords may make some sense with formal designators in other commands, (e.g. SYSDUMP *T/LOCK), only the :FILE command will exclude them. See the MPE Commands Reference Manual (30000-90009) for more information.

Security Increased for File Lockwords

Due to the problem of file lockwords being displayed in job output listings, security has been strengthened to prevent unauthorized persons from seeing the file lockwords. This has been accomplished through the suppression of data following a slash in any job record. All alphanumeric data following a slash will be evaluated to see if it resembles a lockword. If so, it will not be echoed until a special character is reached. Data that resembles a lockword refers to the following criteria:

- o The string begins with an alpha character (not numeric, and not a special character).

- o The length of the data string following the slash until the next special characters is eight (8) character or less.

- o The special character delimiting the data string is a period (.), comma (,), semicolon (;), blank, or carriage return.

When a special character is found, that character and all characters following it will be echoed until the end of the record is reached or another slash is found. If another slash is found, the slash will also be echoed.

Example of input and output for valid lockword data ("LOCKWORD" suppressed):

```
     Input record:
          :BUILD JAN/LOCKWORD.PUB.OSE
     Output record:
          :BUILD JAN/.PUB.OSE
```

Example of data string resembling lockword ("STRING" suppressed):

```
     Input record:
          :COMMENT THIS IS A COMMENT DATA/STRING TEST.
     Output record:
          :COMMENT THIS IS A COMMENT DATA/TEST.
```

Examples of data strings not suppressed:

```
     Non-alpha characters:
          :TELLOP THE TEST BEGAN ON 9/23/82.
          :COMMENT THIS IS THE 2ND/3RD JOB STREAM.
```

String greater than 8 characters:

```
:TELLOP STREAM SUPPRESSION/CONCATENATION TESTS
```

Invalid delimiters at the end of data string:

```
:TELLOP STREAM FIRST/SECOND/THIRD/ TESTS.
```

NOTE

When echoing job records to $STDLIST (typically a line printer), MPE does not distinguish one data record from another record, i.e., no distinction is made between a COMMENT statement and an MPE job command. When it is necessary to display data following a slash that would look like a lockword (meets the above criteria),a blank should be placed following the slash and preceeding the data string. In this case the data string will be echoed. For example:

```
:COMMENT STARTED TESTA/ TESTB.PUB.OSE HERE.
```

:SHOWME Command

The :SHOWME command now reports CPU usage consistently, particularly when :SHOWME is executed programmatically (through the COMMAND intrinsic). Jobs and sessions do not accumulate CPU seconds continuously; rather, they accumulate CPU seconds by updating counts on process termination. The :SHOWME command calculates the CPU seconds by adding the current process's local CPU usage to the accumulated total of all terminated processes. Thus, a programmatic :SHOWME's CPU usage would rarely agree with a :SHOWME command executed in BREAK, since in each case the current process is different.

As of the C/D.01.00 release of MPE, programmatic execution of the :SHOWME command will clearly report that the user is in a program (as opposed to IN BREAK or NOT IN BREAK), and the command will only report the program's CPU usage. The following is an example of the new :SHOWME output when executed in the EDITOR:

```
/:SHOWME
USER: #S1260,JON.OSE,COHEN    (IN PROGRAM)
MPE VERSION:HP32033
CURRENT: FRI, AUG 20, 1982,  9:51 AM
LOGON:   FRI, AUG 20, 1982,  6:43 AM
PROGRAM'S CPU SECS:  15        CONNECT MINUTES: 188
$STDIN LDEV: 34           $STDLIST LDEV: 34
```

Note the differences in the USER line and in the CPU line. Note also that if the COMMAND intrinsic caller is part of a process-handling program, only the calling node's CPU usage will be reported.

COMMAND Intrinsic Now Returns Warnings

The COMMAND intrinsic has been modified to return warnings resulting from improper execution of MPE commands. When a warning is returned through the COMMAND intrinsic, the condition code will now be set to CCE. The error number, however, will be set to a negative value corresponding to the warning number.

When using the COMMAND intrinsic you should first check the condition code. A CCG indicates that execution of the specified command has resulted in an error and the command request has been denied. A CCL indicates that an invalid command was specified by the COMMAND intrinsic. If a CCE is returned, you should further check the error number to determine its value: if a zero is returned the request was granted, and if a negative value is returned a warning has occurred.

SYSDUMP Enhanced

SYSDUMP has been enhanced to call the new version of STORE (described in detail in a separate article). As a result of this change, SYSDUMP will have increased user friendliness, better messages, and higher error tolerance. The new STORE is a program file located in STORE.PUB.SYS, whereas the old version was a group of procedures in the system SL. The ability to turn error recovery on and off has been added. The ability to use indirect files has also been added, as well as fileset exclusion capabilities. This article will explain how to invoke these features while using SYSDUMP.

With the new implementation, the user can pass any of the STORE options to SYSDUMP. Now the user can use the FILES= option to raise the number of files that can be stored. To use any of the STORE options, it is necessary to add them to the response to the "ENTER DUMP FILE SUBSET(S)?" request. For example:

```
     ENTER DUMP FILE SUBSET(S)   @.@.@; FILES=20000
```

Another STORE option that can be useful to change is ONERROR=. By default SYSDUMP runs with ONERROR=REDO, but the user can change this by using the following:

```
     ENTER DUMP FILE SUBSET(S)  @.@.@; ONERROR=QUIT
```

Two new fileset descriptor extentions have been added to STORE. These are Indirect Files and Fileset Exclusion. Both of these capabilities are available in SYSDUMP. These can be used in the same way that they are in STORE. For example, to store files in group PUB.SYS first, enter:

```
     ENTER DUMP FILE SUBSET(S)  @.PUB.SYS, @.@.@-@.PUB.SYS
```

To store all files specified in file INDIR, enter:

```
     ENTER DUMP FILE SUBSET(S)   !INDIR
```

SYSDUMP has a new limit of 240 characters in which to specify dump file subsets. Therefore, if you need more than 240 characters to describe your fileset, you must use indirect files.

NOTE

> If you are running SYSDUMP in a job stream and you want to use indirect files, then you must put a blank in column one, to avoid getting an end-of-the-file on $STDIN.

What :SHOWJOB's EXEC Really Means

:SHOWJOB reports a job or session as being in the "EXEC" state when it is actually executing. However, before execution of the first command begins, the jobs/sessions perform certain initializations, such as prompting for passwords not supplied in the :HELLO command. With the C/D.01.00 release of MPE, :SHOWJOB will report a job/session as being in the "EXEC*" state when it is initializing. The state will change to "EXEC" when the initialization is complete.

User Logging

In past versions of MPE, contrary to documentation, the user logging intrinsics would not perform any capability checking. With this release, a user calling one of these intrinsics will get an error status of 7 (illegal capability) returned via the STATUS parameter if he does not have either LG (logging) or OP (system supervisor) capabilities.

:STORE Command

The new version of STORE has an error recovery capability that allows users to repeat just the reel that had an error instead of the entire STORE operation. When this feature is on, a file that has been locked by STORE will remain locked until after the reel on which that file was STOREd has been completed. This is necessary because STORE must be able to restart the reel if an error occurs. If error recovery is not being used then files will be unlocked as they are stored. In the past, all files have been unlocked as they are STOREd.

FOPEN Intrinsic

FOPEN will now return a FSERR54 (invalid FILE reference) if the *formaldesignator* parameter ends in a period or slash as stated in the MPE Intrinsics Reference Manual (30000-90010). In the past, FOPEN would accept any non-alphanumeric character.

New JOB OVERLOAD Messages

Two new JOB OVERLOAD console messages have been added to the C/D.01.00 release of MPE:

```
JOB OVERLOAD TYPE 6, LDEV #ldev ($STDLIST not allocated)

JOB OVERLOAD TYPE 7, LDEV #ldev ($STDIN not allocated)
```

If a session launch fails because $STDLIST or $STDIN cannot be allocated, the initiating user and the console operator are both informed that the initiation attempt was aborted. If the job launch fails, only the console is informed.

Refer to Appendix A of the Console Operator's Guide (32002-90004) for more information.

Change in Security for System Log Files

The C/D.01.00 release of MPE incorporates changes which will improve the security of closed system logging files.

Logging files are closed when they become full or when a :SWITCHLOG command is executed. Beginning with the C/D.01.00 MIT release of MPE, all file access (i.e. READ, WRITE, LOCK, EXECUTE, APPEND) to system logging files will be allowed to the file's creator only. Previously these files allowed anyone to access them. Since logging files are created by MANAGER.SYS, this means that the closed files will be accessible to that user only.

New System Failure 315

In recent months, the MPE lab has seen some dumps from system hangs that could not be explained. Typically, the hang was caused by a SIR contention, which on the surface looked like a process had locked some SIRs in the wrong order. Closer examination revealed that the user was executing some simple command in his command interpreter, but tracing through the execution of the specified command failed to account for the locking of all the SIRs owned by that command interpreter. Apparently what had happened was that some previous command execution failed to release all the SIRs acquired. This, of course, is an MPE error, but by the time the hang occurred, there was no history of which command caused the error situation.

In order to catch this problem, a new system failure has been added to this release of MPE. A check was placed in the command interpreter which results in a System Failure 315 if a user is holding a SIR after execution of a command. In the event that a System Failure 315 occurs on your system, take a memory dump and WARMSTART the system. Send the hardcopy memory dump and any other supporting information, along with a Service Request, to your local HP Service Representative.

VINIT Enhanced for Fast Disc-to-Disc Backup

by Slawek K. Ilnicki and Adrienne Frost, Computer Systems Division

The VINIT COPY command has been improved to provide enhanced support of disc-to-disc "image" backup. First, by using two buffers, the performance of the VINIT COPY command has been improved. This feature will be of particular interest to users with large discs. Second, by dynamically reassigning defective tracks, the VINIT COPY command will no longer interrupt the backup process for this purpose. In addition, a VINIT VERIFY command has been introduced for checking the readability of data previously written on disc or cartridge tape media.

The enhanced VINIT COPY command and new VINIT VERIFY command support any type of disc volume, i.e. private volumes, system discs, serial discs and foreign discs. The earlier restriction remains that the source and destination discs must be of the same type.

The VINIT COPY function may be used as a private or system volume set image backup facility, but the integrity of the set can only be guaranteed if all volumes are copied simultaneously. The contents of the volume set must NOT be modified before the full backup is complete.

The VINIT COPY and VERIFY commands for private volumes, serial discs and system discs operate only on the used portion of the medium. If you wish to backup or verify the entire medium then you must first scratch the volume (VINIT SCRATCH command). Scratch volumes and foreign discs are copied or verified entirely.

VINIT functions are usually executed interactively, but all except the DOWN command can be performed in batch mode if VINIT if invoked as follows:

```
RUN PVINIT.PUB.SYS
```

COPY Command

The VINIT COPY command will allow the disc backup to continue even if the destination disc has defective or deleted sectors or tracks. Now, VINIT copy will reassign or spare defective or deleted tracks before the backup begins. Optionally, the COPY command will also allow verification of the copied image to ensure that it is readable. If unreadable data is encountered, a message will be sent to $STDLIST reporting which file, and which sectors/tracks within the file are incorrect.

Additional features of the COPY command are:

o Disc-to-disc backup performance has been improved by introducing two internal buffers of 64 Kbytes each. The two buffers allow I/O overlap. To take full advantage of the overlap, configure the discs on two separate GICs. For the small system configuration, there is an option to change the buffers' sizes.

o The COPY command is breakable but not abortable during the verifying process on HP 7911/12 and HP 7933 discs. If the backup process is aborted, the destination disc will be left with a scratch label (indicating a scratched volume).

o Discs from the system domain can only be copied in a restricted environment. This environment is one user per system with no temporary files.

The syntax of the COPY command is:

COPY *ldev1,ldev2* [;GEN=*genindex*] [;VER] [;BUF=*bsize*]

where:

ldev1 is the logical device number of the source disc.

ldev2 is the logical device number of the destination disc.

genindex is the generation index of the volume. Value: 0 to 32767.

VER instructs verify and correct bad data by recopying that portion of the data which appeared bad on the destination disc.

bsize is the internal buffer size in Kwords. Range: 1 to 32; default: 32.

Only two combinations of GEN, VER, and BUF can be specified in one command.

VERIFY Command

The VERIFY command checks the readability of the data on a disc or HP 7911/12 tape cartridge. For mounted private volumes or configured system discs, any files in which unreadable data is encountered are reported to the user to via a message sent to $STDLIST. This command is breakable also, but it is not abortable for HP 7911/12 and HP 7933 devices.

The syntax for the VERIFY command is:

VERIFY *ldev* [;DTT]

where:

ldev is the logical device number of the device to be verified.

DTT instructs to verify only suspect sectors or tracks from the Defective Sector/Track Table.

NOTE: If verifying with the device UP, a warning will be issued, but the verify will continue. Be aware that the verify is invalid if the contents of the disc are changed while the verify is executing.

Performance

Performance has been substantially improved for the HP 7933 discs. When the VINIT COPY command is executed without the verify option, backup time for a 404 Mbyte HP 7933 is approximately:

o 12 minutes, using two GICs and 64K buffers.

o 15 minutes, using two GICs and 32K buffers.

o 21 minutes, using one GIC and 64K buffers.

o 25 minutes, using one GIC and 32K buffers.

For HP7920 discs, the copy procedure is approximately:

o 3 minutes, using two GICs and 32K buffers.

o 4 minutes, using one GIC and 32K buffers.

Error Messages

o **LDEV *x* NOT DOWNED (PVERR 105)** - The devices (source and destination) must be downed during the copy.

o **UNABLE TO REASSIGN TRACKS (PVERR 225)** - No spare tracks are available.

o **WARNING: System logging disabled (PVWARN 226)** - System logging will be disabled during the copy of the configured system disc.

o **WARNING: System logging enabled (PVWARN 227)** -

o **WARNING: LDEV *x* - Volume not logically mounted (PVWARN 228)** - The volume must be mounted in order to detect files with lost data.

o **WARNING: Unable to report files with bad or lost data LDEV *x* - Volume not logically mounted (PVWARN 229)**

o **UNREADABLE OR LOST DATA IN SECTORS *w* - *x* (TRACK: CYL *y* HEAD *y*)**

o **WARNING: Suspect or deleted sectors/tracks on LDEV *x* Beginning spare or reassign (PVWARN 231)**

o **UNABLE TO COPY SYSTEM DISC - MORE THAN ONE USER/OPEN TEMPORARY FILES/OPEN $OLDPASS (PVERR 234)**

DISKED2 as an Octal File Editor

by Slawek K. Ilnicki and Adrienne Frost, Computer Systems Division

The functionality of the DISKED2 utility has been extended to include the concept of an Octal File Editor. The Octal File Editor will allow the user to display or modify any sector of a temporary or permanent file. It is not necessary for the user to have System Manager (SM) capability to do this. The user who does have SM capability can still use DISKED2 in the original mode, i.e., as a Disc Editor.

DUMP Command

The DISKED2 "DUMP" command has been enhanced to allow the user to display any sectors of a file. To do this, the user must minimally have READ access to the file. Also, the new option "ALL" has been added to allow a file to be displayed up to 65536 sectors or until end-of-data.

The new syntax is as follows:

```
     DUMP ALL
or
     DUMP ALL,A   (if ASCII only)
```

MODIFY Command

To modify the contents of a file, the user must issue a MODIFY command. Exclusive WRITE access to the file is required and only the user with SM capability is able to modify the file label (sector 0 of the file). If the user does not have exclusive WRITE access and tries to perform a MODIFY command, the function is terminated and the following warning is displayed:

```
     WRITE ACCESS IS REQUIRED
```

In addition, the MODIFY command has been enhanced to allow users to input values as ASCII or decimal. The format for these values is as follows:

```
     'xx' or 'xx  - modifies two bytes (ASCII)
     'x'  or 'x   - modifies left byte (ASCII)
     #yyyyy       - decimal value (integer)
```

FILE Command

In order to use the DUMP and MODIFY commands successfully, the user must first open the file. The new DISKED2 "FILE" command allows the user to open a file, closing the currently open file, if any. If the user issues the DUMP or MODIFY command without first specifying a file, he will be prompted for the file name, as follows:

```
     FILE?
```

By entering a valid file name, that specific file will be opened. The command can be terminated by entering a carriage return (RETURN) instead of a file name.

The FILE command has the following syntax:

```
    F[ILE] filename
```

This command does not allow file equations. The following message appears when you try to use a file equation:

```
    FILE EQUATION DISALLOWED
```

The user with SM capability may switch from File Editor to Disc Editor mode by entering the DISC command.

Also introduced in DISKED2 is the Control-Y function to allow the termination of the DUMP and MODIFY command. Control is returned to the DISKED2 command level.

Error Messages

The following additional warnings and error messages appear in the enhanced DISKED2:

o ****FILE EQUATION DISALLOWED**

o ****WRITE ACCESS IS REQUIRED**

o **SECURITY VIOLATION (FSERR 93)**

o **NONEXISTENT PERMANENT FILE (FSERR 52)**

o ****SYS. MGR CAPABILITY IS REQUIRED**

o ****SYS. MGR CAPABILITY IS REQUIRED TO MODIFY FILE LABEL**

MPE Stack Size Requirements

by Kim Rogers, Computer Systems Division

Since the C.00.02 release of MPE, the stack size requirements for some of the file system intrinsics have increased. For programs on the C.00.02 release that are currently running at or near their current stack limits, it may be necessary to increase the initial stack size and/or *maxdata* values in order to successfully run these programs with the C/D.01.00 release. An alternative would be to reconfigure either the maximum or standard stack size values to be larger to accommodate the new MPE requirements.

The following table compares approximate stack size requirements between the C.00.02 ("OLD") and C/D.01.00 ("C/D") releases. These estimates may help in determining any potential for problems.

MPE Stack Size Comparison

	New disc file;save		Old disc file		Dev = LP no ENV		Dev=EPOC ENV	
FILE SYSTEM INTRINSICS	OLD	C/D	OLD	C/D	OLD	C/D	OLD	C/D
FOPEN	893	1054	902	1054	1040	1089	2113	2271
PRINTFILEINFO	532	578	532	578	390	442	390	441
FFILEINFO	364	387	364	387	380	406	380	406
FGETINFO	214	224	214	224	229	243	229	243
FWRITE	294	344	294	344	166	226	166	226
FREADDIR	191	229	191	229	-	-	-	-
FWRITEDIR	407	404	299	350	-	-	-	-
FCONTROL	188	197	188	197	-	-	-	-
FREAD	299	384	299	348	-	-	-	-
FCLOSE	474	925	444	925	509	925	509	925

NOTE: The values shown for disc files in the C.00.02 columns are the same for MPE version C.00.01. All values shown in the C/D.01.00 column are the same for MPE versions C.00.08 and C.00.20/D.00.20. All values are in words.

On C/DOD.01.00, an FOPEN of a disc file on a private volume will take approximately 1412 words of stack. On an FCLOSE of a disc file on a private volume will take approximately 1214 words of stack space. All other values shown in the above table remain unchanged.

Program File Preparation Changes

by Jessy J. Hsu, Computer System Division

Multiple-Extent Program File

A program file may include symbolic items and PMAP information according to the user's request. (Users will either ask for symbolic debugging, or simply for inclusion of the PMAP information). Thus the program file size may become very large. To avoid problems that frequently occur upon allocation of disc space for a very large extent, the SEGMENTER (version A.01.08, on the C/D.01.0C release of MPE), will create multiple-extent program files, providing that all the code is still within one extent.

When a multiple-extent program file is created, it is closed as uncrunched. This is because a copy from a crunched file using FCOPY, or other file copying subsystem, may result in a different extent size, thus creating a program file with code crossing an extent boundary. Such a program file will be rejected by the LOADER as an invalid program file.

Preparation of an USL File into an Existing Program File

Do you remember the frustration of getting "ERROR 84 UNEXPECTED I/O ERROR" while you tried to prepare an USL file into an existing program file which was not big enough? This problem has been solved! The SEGMENTER now will always try to prepare a program into a new file and close this file as a temporary or permanent file, as specified. (Of course the existing file will be purged). This new scheme allows changing file characteristics such as file size and extent size. The program file size will be expanded or contracted depending upon the file space needed. This scheme also ensures that all the code is within one extent by specifying file attributes for the new file.

The user must have Save File (SF) capability to the group and account where the existing program file resides. If not, or if disc space has been exhausted when the SEGMENTER tries to open a new file, the SEGMENTER will use the existing file. In this case, two consequences may occur. First, ERROR 84 may occur if the file is too small. Second, the code may cross an extent boundary if the existing file has more than one extent and the extent size is too small.

User Logging Tape Power Failure Recovery

by Kim Rogers, Computer Systems Division

In systems running versions of MPE, previous to C.01.00/D.01.00, if a power failure occurred during a "quiet" period (i.e. between writes to the logging tape), the usual power failure recovery routine was not always sufficient to guarantee recovery of the disc logging buffer. As a result, the Console Operator's Guide stated that it was safer to HALT the system and then to WARMSTART (to invoke the warmstart recovery routine) after the power failure. This procedure is no longer necessary.

Now, in systems running version C.00.08 or later, if the power failure occurs during a "quiet" period, the I/O System will record this fact. When the next write is performed to the tape, the I/O System will return "power failed" status. This, in turn, will cause User Logging to invoke its power failure recovery routine. Upon detection of the power failure, a message asking the operator to place the tape drive back on-line will appear on the console:

> **Power fail detected on ldev #*n*. Please put tape drive back on-line (hit LOAD,RESET,ON-LINE) (ulogmsg 32)**

As soon as the operator places the tape drive on-line, the tape will be rewound to the load point and then forward-spaced to the point where the power failure occurred. At this point, User Logging will continue writing to tape with no interruption to the user's application. (Note: The operator should not place the tape drive back on-line until the message appears).

HP 3000 Asynchronous Serial I/O Printer Support

by Steve Couch, Information Networks Division

The C/D.01.00 release of MPE has been enhanced, to provide increased spooled printer support.

HP 3000 serial I/O printer support can be divided into three groups as follows:

o HP 2631B support as TERMINAL TYPE 19.

o Enhanced HP 2631B support.

o Generic asynchronous serial I/O printer support.

HP 2631B Support As TERMINAL TYPE 19

This support for the HP 2631B was introduced on the 2028 release of MPE. For detailed information one should read the article "INTRODUCING THE 2631B REMOTE SPOOLED PRINTER" by Terry Ishida, in issue 25 of the HP COMMUNICATOR.

Enhanced HP 2631B Support

There were two enhancements to HP 2631B support added to the the ADCC and ATP drivers. These enhancements are not available for the ATC driver.

o The first enhancement will allow the use of 8-bit data with the HP 2631B. This enhancement was introduced specifically for the Katakana (Japanese) character set, but any 8-bit character set may be used. The I/O configuration for the use of this enhancement is the same as that used for a standard HP 2631B, except that TERMINAL TYPE 20 should be used instead of TERMINAL TYPE 19.

In addition, the following printer dip switches should be set differently than was originally specified for the standard 7-bit data HP 2631B switch settings:

S2-2	OPEN	NO PARITY
S2-3	OPEN	NO PARITY
S4-7	OPEN	DO NOT STRIP EIGHTH BIT
S5-7	CLOSED	EIGHT BIT DATA COMMUNICATION

Since all eight bits are used for data there is no way to check data integrity. Thus data transfer errors may occur and the user will not be aware of this condition. For this reason, the use of 8-bit mode over MODEMs is not suggested. It should be noted that not all HP 2631B were shipped with printed circuit assemblies (PCAs) which can be used for 8-bit data transfer. All HP 2631B's with a serial I/O PCA containing two sets of dip switches have this option.

- The second enhancement added for the HP 2631B is intended to allow the user to safely embed escape sequences in the data stream. If the system embeds a printer status request in the middle of the user's escape sequence, the printer will not process the escape sequence correctly. This problem can be avoided by the use of the new TERMINAL TYPES 21 and 22. These TERMINAL TYPES do not request status when an XOFF is received, or at the end of a write. Status will only be requested after the driver adds carriage control information to the data. Instead of requesting status when an XOFF is received, the driver will now start a 60-second timer. If the timer expires, a "LDEV XXX NOT READY" message is printed on the console. As a result, the messages are not as timely or as detailed as with TERMINAL TYPE 19. The more detailed messages are issued whenever the status request can be performed safely. To use this enhancement, specify TERMINAL TYPE 21 for 7-bit data and TERMINAL TYPE 22 for 8-bit data.

Generic Asynchronous Serial I/O Printer Support

Limited support is offered for other TERMINAL TYPES. This facility is offered to allow the spooling of other HP serial I/O printers that are not HP 2631Bs, but it may also be used for non-HP devices. It is important to understand that HP cannot guarantee the successful operation of any specific device under this limited support configuration. It should also be noted that even HP serial I/O printers configured as terminal types other than 19-22 are limited in data integrity and format control. Such devices may be spooled when configured as TYPE=32, SUBTYPE=14 or 15 and any supported TERMINAL TYPE. These TERMINAL TYPES will continue to operate as previously documented. The most significant differences that exist when a device is configured as type 32 rather than 16 are as follows:

- The device may be spooled when configured as TYPE 32.

- Only functions that are valid for printers will be allowed when a device is configured as TYPE 32. This precludes the programmatic changing of SPEED, PARITY, TERMINAL TYPE, and DATA SIZE.

- On the ADCC and ATP a "form feed" will be issued at the end of each spool file or when the user closes the device if the device is not already at "top of form". The ATC driver will continue to issue a "CR,LF" when a device is opened or closed and the device was not left on a "new line".

- On the ATP redundant form feeds which are specified in the the carriage control are supressed.

- o When a user does not have the device spooled he will see the following differences:

    1. On the ATP a write request will not be completed until all data and carriage control associated with the request has been transfered to the device.

    2. On the ATC and ADCC a write request will be completed when the data and carriage control have been moved to a terminal buffer.

Some of the limitations which a user will notice between "generic asynchronous printer support" and standard "system printers" are that:

- o There is no support of automatic page eject.

- o There is no support of vertical format control, (VFC 1-16).

- o There is no guarantee of data integrity and no reporting of data transfer errors.

- o There is no initialization of the device. If the last user of the device left it in a "strange state" (i.e. alternate character set, large characters, etc), it will remain there until reset by another user.

It is recommended that HP 2601s and similar non-ENQ/ACK devices be configured as TERMINAL TYPE 18.

The ADCC and ATP drivers will send 8-bit data to all "generic asynchronous printers" and will expect to receive flow control characters (XON/XOFF), with the parity bit set to 0. The ATC driver will send 7-bit data to these printers with the parity bit set to odd parity if the configured TERMINAL TYPE uses 7-bit data. If the TERMINAL TYPE on an ATC port specifies 8-bit data, the ATC driver will act like the ATP and ADCC drivers. The ATC driver will ignore parity on flow control characters if the TERMINAL TYPE does not use 8-bit data.

Configurations

Printers attached to the terminal I/O controllers are supported when configured as: o TYPE=32, SUBTYPE=14 (directly-connected), and TERMINAL TYPE is 19, 20, 21, or 22.

o TYPE=32, SUBTYPE=15 (U.S. MODEMs), and TERMINAL TYPE is 19 or 21.

TERMINAL TYPES 20, 21, and 22 are not implemented on the ATC driver. The driver name will depend upon the type of terminal controller:

- o For all devices attached to an ADCC (HP 3000 Series 30,33,40, or 44), the driver name must be HIOTERM0.

- o For all devices attached to an ATC (HP 3000 Series II or III), the driver name must be IOTERM0.

o For all TYPE=32 devices connected to an ATP (HP 3000 Series 64), the driver name must be HIOASLP0.

All devices configured as TYPE=32 will operate as spooled devices under the MPE spooler. Such devices may be configured as "initially spooled", or they may be spooled after the system is up.

Users of serial I/O printers should remember that these devices are intended only for light-duty print jobs, and they should not be used for large volumes of data or jobs where absolute data integrity is neccessary. The only serial I/O printer that can detect and report transmission errors is an HP 2631B configured as TERMINAL TYPE=19 or 21.

The only source of data integrity for other serial I/O printers on the HP 3000 is the quality of the electrical connection. When using shielded cables and operating within EIA standards (50 feet for RS-232-C and 4000 feet for RS-422), the signal quality is excellent. Beyond these limits, the quality degrades and data transmission errors become more likely at higher data rates. Serial I/O printers which are connected via a MODEM will be highly vulnerable to transmission errors, primarily because of the noise that exists on phone lines. The HP 39301A Fiber Optic Multiplexer offers excellent noise immunity on connections to 3281 feet (1000 meters).

Announcing VPLUS Support for 2624B Local Form Storage

by Ron Harmar, Information Networks Division

With MPE version B.03.03 of VPLUS/3000 announces local form storage using the forms cache feature of the 2624B terminal. The C/D.01.00 version of VPLUS/3000 also contains several other fixes and enhancements, as well as continuing support for a wide range of terminal operations. This article introduces local form storage on the 2624B, describes the other changes in the new version of VPLUS, and offers new information about the local edits feature, first announced in version A.02.00 of VPLUS/3000.

Questions and Answers

o What is local form storage?

Local Form Storage (LFS) is the VPLUS implementation of special terminal features that permit loading and display of forms in terminal memory. VPLUS provides the user with a wide variety of LFS options, and manages these options through a dialogue of escape sequences and terminal status returns executed on the user's behalf. In a properly designed application, LFS can reduce data communication overhead, permit concurrent form downloading, and provide a faster, more aesthetic method of form display.

o What is actually stored in LFS?

The "form" in local form storage refers to the empty screen design created in FORMSPEC. Any initial values, window messages, or function key labels are not stored (VSHOWFORM adds these after the form is displayed).

o How will LFS help my applications?

A well-designed LFS application can reduce data communication overhead. LFS requires at least as many terminal writes as a non-LFS form transfer, plus another FWRITE for each downloading operation. The savings occurs in the number of characters transmitted over datacomm, since an LFS trigger is an escape sequence (about 10 characters for the 2624B and about 60 for the 2626A) while a large VPLUS form may contain thousands of characters. To achieve this savings, an application should maximize the triggers and minimize the downloads.

In a point-to-point environment, the LOOK'AHEAD word of COMAREA can be set to enable concurrent form downloading during VREADFIELDS. If LOOK'AHEAD is enabled, the next form in sequence is loaded while the operator types data in the displayed form.

An LFS display is generally faster than transmitting the form over datacomm, especially on heavily loaded systems. A form displayed from LFS also has the aesthetic advantage of appearing all at once, rather than being painted line by line.

- How do I activate LFS?

  LFS is supported on the 2624B and 2626A terminals. It is enabled by setting FORM'STORE'SIZE* to a value between 1 and 255 prior to VOPENFORMF. (The same range may be used on all terminal types without causing an error.) On terminals other than the 2624B and 2626A, LFS will not be enabled. On the 2626A, the programmer may use any FORM'STORE'SIZE value between 1 and 255, but VPLUS will not exceed the 2626A's capacity of four forms. The 2624B can store up to 255 forms.

- How do I load forms into the terminal?

  VPLUS now provides three ways to load forms into LFS. The VLOADFORMS intrinsic allows the programmer to load one or more forms in a single call, generally at the start of a program or routine. The two other methods permit an individual form to be downloaded as needed. If the LOOK'AHEAD word of COMAREA is set to 0, the next form (as defined in the FORMSPEC Form Menu) will be loaded during VREADFIELDS. In a point-to-point configuration, the downloading operation takes place before the screen is read, while the operator is typing data into the displayed form. In a multipoint environment, the loading occurs after the read.

  With C/D.01.00, setting bit 9 (adding 64) to the SHOWCONTROL word of COMAREA enables preloading during VSHOWFORM. If VSHOWFORM does not find the desired form in terminal memory and SHOWCONTROL bit 9 is set, VSHOWFORM loads the form and displays it from local storage.

  To conserve LFS memory, the look-ahead and preload options will not load a form if another form in the same family is already in storage. If your application requires it, VLOADFORMS may be used to load more than one member of the same form family into LFS.

  VLOADFORMS, LOOK'AHEAD, and SHOWCONTROL bit 9 operate on the 2624B and 2626A only. However, to simplify programming, their use on other terminals does not cause an error.

- How are forms purged from terminal memory?

  The VUNLOADFORM intrinsic allows the programmer to specify a form to be taken out of local storage. Otherwise, the loading options described above will make room for new forms by purging old ones on a least-recently-used basis. If the programmer wants to disable this automatic recycling process, the LOOK'AHEAD word of COMAREA should be set to 1 and bit 9 of SHOWCONTROL should be reset to 0. LFS is completely cleared by calling VOPENTERM, using

*Refer to the VPLUS Reference Manual (32009-90001) for more information about the FORM'STORE'SIZE, LOOK'AHEAD, and SHOWCONTROL words of COMAREA.

$REFRESH, or powering-down the terminal. On a 2626A, appending to a form stored in the terminal removes it from LFS, since the appending operation works directly on the stored form.

o How many forms do the terminals hold?

In the 2626A, the form is stored in its displayed state, complete with protected and unprotected fields, headings, and enhancements. Each form is given a 26-line "workspace" in the 119-line display memory of the 2626A, so the maximum number of forms is fixed at four (a form larger than 24 lines requires a larger workspace, and may reduce the number of forms that can be stored). In contrast, the 2624B stores a VPLUS form in its FORMSPEC-compiled state, as a string of ASCII characters and escape sequences. The data compression feature of the 2624B further reduces the memory requirement of each form so that the maximum number of forms that can be loaded depends on the size and design of the forms and on the amount of memory in the terminal. The absolute maximum of 255 forms has been achieved using extremely small forms, but the practical limit is lower.

o Does the 2624B always hold more forms than the 2626A?

Consider the extreme case of a form with 128 ten-character fields, which would produce a compiled screen of about 2,900 bytes. VPLUS can fit four of these in the 2626A, and twelve in a 2624B with 32K bytes of memory.

A more typical case is a form like the FORMSPEC Main Menu, which has a compiled screen of about 660 bytes. Again, the 2626A holds four of these. The 2624B with 32K bytes of memory can store 43.

The best way to determine whether your own forms are more like the extreme case, the typical case, or the best case (of which 255 can be stored) is to experiment with VLOADFORMS on a 2624B with the desired memory option.

o How long does it take to load a form?

Since form loading is a datacomm transfer, the loading speed depends on the baud rate and is roughly the same for both LFS terminals.

o How fast are the stored forms displayed?

When LFS is used, both terminals display the form all at once, rather than painting it line by line. Since the form on the 2626A is stored in displayable form, the display operation simply consists of switching the desired form to the active window. Displaying a form stored in the 2624B takes somewhat longer, because the form must be copied from forms cache to the display area. During this copy operation (which takes place with video display turned off), the 2624B executes the escape sequences contained in the compressed form and creates the displayable form. When VPLUS turns the video display back on, a fresh copy of the desired form appears on the screen.

The advantage of the 2624B method is that the displayed form is always a clean copy, ready to receive data. In contrast, the 2626A stored form must be continually "repaired" by VPLUS, because any error enhancements, field type changes, or other modifications are made directly on the stored original. The disadvantage for the 2624B is that its copy time depends on the number of characters in the compressed form (including display enhancements, local edits, and cursor positioning sequences) while forms stored in the 2626A are switched at a constant speed (roughly one refresh cycle) regardless of their size or complexity.

On an idle system running at 2400 baud, the 128-field form described above takes about 13 seconds to be painted in a non-LFS display. When displayed from LFS, it appears in about 4 seconds on the 2624B and almost instantaneously on the 2626A. The advantage of the LFS display is even more apparent on heavily loaded systems. Once the LFS escape sequence reaches the terminal, the stored form is displayed locally without having to compete for datacomm.

o When should I use LFS?

The advantages of LFS in general depend on system characteristics and application requirements. Good candidates for LFS include remote terminal applications and systems running at slower baud rates, menu-driven programs in which an operator-selected task can be preceded by downloading the forms needed for the task, and any block-mode environment where line-loading is critical.

The advantages of each LFS-supporting terminal may be summarized as: fresh-copy display and generally more compact storage on the 2624B, and generally faster form switching on the 2626A. The 2624B also provides better support for appended forms.

o Do my applications have to change for the 2624B?

Applications that already use local form storage on the 2626A will automatically enable the same LFS options on the 2624B without any programming changes. However, the application designer may wish to change the settings for the FORM'STORE'SIZE word of COMAREA and the NUMOFFORMS parameter in VLOADFORMS to reflect the 2624B maximum of 255 forms. When setting FORM'STORE'SIZE, remember that VOPENFORMF takes your FORM'STORE'SIZE value as the number of 15-word LFS directory entries to build in the COMAREA extension. Since the 2624B capacity is usually less than 255 forms, routinely setting FORM'STORE'SIZE to 255 may waste thousands of words of stack space.

Terminal-dependent applications may now test COMAREA word 58 (relative to 0) for a value of 13, indicating a 2624B terminal. In previous versions, VPLUS supported the 2624B as a non-LFS 2624A, with an identifier of 9.

The programmer may also wish to enable the new SHOWCONTROL pre-load option, described below. This option may be used on either LFS terminal type.

Other Changes in the C/D.01.00 Version of VPLUS

Along with support for LFS on the 2624B, the C/D.01.00 version of VPLUS makes several changes in the forms file and intrinsics. These changes are previewed below, and will be documented in a new edition of the VPLUS Reference Manual (32209-90001, Fourth Edition).

o Changes in the Forms File:

Forms files created prior to the C/D.01.00 version of VPLUS/3000 will run on the new version, with certain exceptions. A forms files created prior to B.03.03 will automatically invoke the new Adjust Menu in FORMSPEC until the file is recompiled under the new version. The Adjust Menu controls a cross-checking procedure made necessary by changes in certain forms file record layouts. When this menu appears, the user may continue the cross-checking operation, select a different forms file, or exit from FORMSPEC. Once the forms file is cross-checked, it must be recompiled and will no longer run on earlier versions of VPLUS. For this reason, you may wish to keep a backup copy of your forms files before converting them. Forms files created or recompiled with the new version of FORMSPEC may now contain up to 32767 forms, versus the previous limit of 255. Because of this change, any forms file that is accessed by calls to VGETFORMINFO and has not been recompiled since the Athena MIT (A.01.01) should be recompiled using the new version of FORMSPEC.

Forms files created or converted under the new version will keep forms in alphabetical order (in ASCII collating sequence) by form name. Form designers accustomed to having their forms kept in order of creation may notice the following operational changes:

1. By default, the head form has always been the first form in the file. In previous versions, this meant the first form created. In the new version, the default head form is the first form in alphabetical order. As always, the forms designer may specify a different head form name in the Globals Menu.

2. The "PREV FORM" (f1) and "NEXT FORM" (f2) keys in FORMSPEC will get the next or previous form in alphabetical order. The "HEAD FORM" (f1) key in ENTRY will get the first form in alphabetical order, unless a different head form was specified in FORMSPEC.

3. The forms in a FORMSPEC listing will be alphabetized.

o Changes in VSHOWFORM and SHOWCONTROL

Two new bits have been defined in the SHOWCONTROL field of COMAREA (word 33, relative to 0). As described above, bit 9 of SHOWCONTROL is now used as a form preload indicator. If VSHOWFORM does not find the desired form in local form storage and the programmer has set bit 9 to 1, the form is first loaded into and then displayed from terminal memory.

SHOWCONTROL bit 10 now controls keyboard unlocking during VSHOWFORM. Setting bit 10 to 1 signals VSHOWFORM to leave the keyboard locked after displaying a form.

In previous versions of VPLUS, one undocumented effect of VSHOWFORM was to set SHOWCONTROL to 0 after each call. Starting with the C/D.01.00 version, VSHOWFORM will leave this word unchanged so that the programmer may set the desired SHOWCONTROL bits once for the entire run of a program. Applications that have relied on VSHOWFORM to reset SHOWCONTROL will now have to set this word explicitly.

Since C-Delta (B.02.00), VSHOWFORM has rewritten the data buffer to the screen whenever a form is redisplayed after a function-key read. The rewrite occurs on all terminal types, though its purpose is to reset the modified data tags of the 2624B. An unwanted effect of this rewrite is that any values typed into the form before the function key was pressed are overwritten with the old data buffer contents. With C/D.01.00 of VPLUS/3000 the modified data tags are still reset, but the rewrite approach is no longer used. An example of this can be seen in FORMSPEC when the user types data in the Field Menu and presses the f3 key twice. In previous versions, this would replace the user inputs with default values; in the new release, the inputs are not disturbed.

o Local Edits

VPLUS support for local edits on the 2624B* terminal was first announced in version A.02.00. The local edit feature has not changed in C/D.01.00 of VPLUS/3000 but the following material is offered as a supplement to the current documentation.

Local edits provide high level access to the format mode features of the 2624B terminal. By defining a CONFIG phase in the processing specifications area of the Field Menu and selecting the "HP262X Family" in the Terminal Selection Menu, the VPLUS user may choose appropriate combinations of local edits for any or all fields in a form. The escape sequences needed to enable these edits are embedded by FORMSPEC into the compiled screen design and loaded into the terminal by VSHOWFORM when the form is displayed. In the following tables, the local edits that may be specified in FORMSPEC are shown along with the 2624B menu selections that perform the same functions when the terminal configuration softkeys are used.

o Local Edit Combinations

The use of local edits in VPLUS is governed by the rules of format mode in the 2624B terminal, which means that certain combinations of edits are not allowed. FORMSPEC will flag most of these combinations as errors before they are captured by the 2624B. The validity of other combinations depends on their context, so their use is left to the judgment of the forms designer.

*Unlike local form storage, which is available only on the 2624B, local edits are supported on both the 2624A and 2624B. In this article, all references to local edits apply to both terminals.

Table 1. Local Edits Corresponding to Field Type Selections in the 2624B Edit Checks Menu

VPLUS Local Edit	2624B Menu Selection	If JUSTIFY Is Used
UNRESTRICTED	0. ALL CHARACTERS	Left justify
ALPHABETIC	1. ALPHABETIC	Left justify
UPSHIFT	2. AUTO UPSHIFT	Left justify
ALPHANUMERIC	3. ALPHANUMERIC	Left justify
INTEGER	4. INTEGER	Right justify
SIGN_DEC	5. SIGNED DECIMAL	Right justify
IMP_DEC	6. IMPLIED DECIMAL	Ignore
CONSTANT	7. CONSTANT	Ignore
INTEGER_FILL	8. INTEGER/FILL	Ignore
SIGN_DEC_FILL	9. SIGNED DECIMAL/FILL	Ignore
IMP_DEC_FILL	10. IMPLIED DECIMAL/FILL	Ignore

Table 2. Local Edits Corresponding to Field Attribute Selections in the 2624B Edit Checks Menu

VPLUS Local Edit	2624B Menu Selection	Default
REQUIRED	REQUIRED	OPTIONAL
JUSTIFY	JUSTIFY	NO JUSTIFY
MUST_FILL	TOTAL FILL	NO TOTAL FILL

For each field, the user may select one edit from table 1 and any combination of edits from table 2, according to the following guidelines:

1. If no selection is made from table 1, the default edit of UNRESTRICTED is used.

2. The default attributes shown in table 2 are invoked by omitting the corresponding local edit from the CONFIG phase (for example, if the REQUIRED attribute does not appear in a LOCALEDITS statement, the field is treated as OPTIONAL by the 2624B).

3. The CONSTANT and REQUIRED edits are incompatible, since the required data cannot be entered in a CONSTANT field.

4. The terminal executes all edits implied by the field type before it checks for total fill. For example, data entered in an IMP_DEC field

is aligned on the decimal point and trailing zeros are added. If these implicit edits fill the field, then the MUST_FILL stipulation is met. The MUST_FILL edit is always met by INTEGER_FILL, SIGN_DEC_FILL, and IMP_DEC_FILL.

5. The effect of the JUSTIFY attribute on each of the 2624B field types is shown in table 1.

Along with any selections from tables 1 and 2, the user may define the number of decimal digits (DEC_DIGITS n) and select a decimal format (DEC_TYPE_EUR or DEC_TYPE_US), as shown in table 3. Unlike the other local edits, the DEC_DIGITS and DEC_TYPE edits are global terminal configurations that remain in effect until reset by another DEC_DIGITS or DEC_TYPE edit. As a result, if more than one field in a form specifies DEC_DIGIT or DEC_TYPE values, only those values defined for the last field in screen order will take effect when the form is displayed. Also, since VOPENTERM and VCLOSETERM do not reconfigure DEC_DIGITS and DEC_TYPE, these settings may carry over from one program to the next unless explicitly reset. In practice, DEC_DIGITS and DEC_TYPE should be set once, near the start of the program. Fields requiring different local decimal configurations should be displayed in separate forms with separate DEC_DIGITS or DEC_TYPE settings.

Table 3. Local Edits Corresponding to Format Mode Selections in the 2624B Terminal Configuration Menu

VPLUS Local Edit	2624B Menu Item	Menu Selection
DEC_DIGITS n	Implied Dec Digits	0 - 9
DEC_TYPE_EUR	Decimal Type	EUR
DEC_TYPE_US	Decimal Type	US

o Transmit-Only

In addition to the edits from tables 1, 2, and 3, the TRANSMIT_ONLY edit may be used to define a field as transmit-only. This edit corresponds to the START XMIT FLD key in the 2624B Aids Key group, and acts as a signal to FORMSPEC to use an ESC { start-of-field delimiter. For more information about transmit-only fields, consult the 2624B Reference Manual (02624-90008).

o Local Edits and Form Families

The optimizations of VSHOWFORM may affect the status of local edits in family forms. Each member of a forms family has its own compiled screen in the forms file, so each may have a different set of local edits embedded in its screen. However, when family forms share a displayed screen at run time, only those edits for the form that first caused the screen to be displayed are used. Therefore, unlike the processing specifications defined in other Field Menu phases, local edits cannot be changed by getting and displaying another family member.

Users familiar with form families may recall that forms in the same family can be designed with different field types and display enhancements, and that VSHOWFORM will make the required updates to the displayed form at run time. However, changing the field type in a displayed form will cause the local edits in the changed field to be lost.

o Local Edits and Other Processing Specifications

FORMSPEC allows the forms designer complete freedom in combining the local edits of the CONFIG phase with edit specifications in other Field Menu phases. Although local edits can only be used in the CONFIG phase, most have counterparts in other edit phases that may be used in addition to or in place of the local edits. In judging the suitability of local edits for a particular application, you should refer to the VPLUS and 2624B Reference Manuals to understand the role of local edits and their relationships to other VPLUS processing specifications (examples of these relationships are are shown in table 4). This is particularly important for applications that must keep data entered on other terminals consistent with data processed by the 2624B edits.

Table 4. Some Local Edits and Their Counterparts in Other Phases of the Field Menu

VPLUS Local Edit	Counterpart Specification
UNRESTRICTED	DType: CHAR
ALPHABETIC	DType: CHAR Processing Specifications: MATCH a*
IMP_DEC (with DEC_DIGITS 2)	Field Name: FIELDX DType: NUM2 Processing Specifications: MATCH d*.[d*] "Illegal for edit type IMP DEC" SET TO FIELDX
CONSTANT	FType: D
INTEGER_FILL	DType: DIG Processing Specifications: JUSTIFY RIGHT FILL LEADING "0"

Summary

The C/D.01.00 version of VPLUS includes: support for local form storage on the 2624B, changes to the forms file allowing a much larger maximum number of forms, and enhancements to the intrinsics, including new SHOWCONTROL options and removal of the data-buffer rewrite after a function-key read. These changes, along with continuing support for a wide range of advanced terminal features, should make the C/D.01.00 version of VPLUS/3000 an attractive upgrade option for all VPLUS users.

Introducing HPToolset

by Tad Olson, Information Networks Division

On June 22, 1982, HP introduced HPToolset, a new productivity software package which broadens the number of solutions available for increasing programmer productivity. It works in conjunction with the latest release of COBOL II (A.00.08) and Segmenter (A.01.07) to provide a uniform programming environment for today's COBOL II programmers on the HP 3000.

Overview of HPToolset

HPToolset is an interactive program development system designed to help the application programmer develop COBOL II programs on the HP 3000 computer. Its simple interface makes it easy for programmers to create, manage, and test COBOL II programs. HPToolset increases programmer productivity and satisfaction by decreasing the time, expense, and complexity of COBOL II program development and maintenance.

To help learn HPToolset, an on-line HELP facility is provided which defines all features, commands, and options. This HELP facility makes it easy to learn HPToolset simply by using it.

Screen-labeled function keys and menus are used for task selection and processing functions. A comprehensive set of commands allows the programmer to move between the creation, translation, and execution stages of program development as well as to execute MPE commands without leaving the HPToolset environment.

HPToolset's Symbolic Debug lets even novice programmers quickly produce error-free application programs without having to work at the primitive level of memory locations. An easy-to-use interface lets users set program breakpoints and display/modify data items with ease.

Features

o Workspace Manager:

One of the ways HPToolset makes programming easier is that it helps the programmer manage the files needed to create COBOL II programs. It does the "bookkeeping" for files and relevant information on each of these files in a WORKSPACE root file. By associating these files with a WORKSPACE, HPToolset maintains a directory of all files used to create and run one program. Files managed by HPToolset's Workspace Manager include:

1. Source Files.
   - -Owned files (files created by the programmer).
   - -Used files (files Owned by another workspace but shared).
   - -Different versions of the same source file.

2. USL file.

3. Program file.

4. Latest compilation listing file.

All source files that are created within HPToolset are in keyed HPToolset internal format and are accessed via the Toolset Access Method or "TSAM".

TSAM provides for:

1. Version Management: Each source file in the Workspace consists of one or more versions which may differ from each other. A version is a logical copy or "snapshot" of the file. Changes made to the file in the form of additions, deletions and updates are kept in versions of the same file. You can control when this "snapshot" of the file is taken through the SETVERSION command.

   Duplication of source code is not included in each version of the source file resulting in better disc utilization. You may have up to 32 active versions per source file.

2. Data Record compression: All data records are compressed on disc resulting in more efficient use of disc space.

3. Optimized Sequential access: Sequential retrieval of data is optimized.

o Full Screen Editor:

Direct screen editing of text using HPToolset's versatile full screen editor saves time through its ease of source code entry and modification. You have full use of the editing and cursor control keys on the terminal keyboard to go to any part of the screen and make changes to your source code. Changes are accomplished by simply retyping over the existing text.

A full range of labeled function keys and commands increase editing capabilities to add, find, change, move, and copy text. The HPToolset editor will set tabs and margins appropriate for the particular language you are using.

Edit options can be set when you create a file or when you convert an ASCII file to TSAM format (defaults are underlined):

-SOURCE LANGUAGE	[COBOL] [OTHER]
-STRUCTURE	[MAIN] [SUBPROGRAM] [$INCLUDE] [OTHER]
-PROGRAM-ID	
-LINELENGTH	[74]
-FILE SIZE	[6000]
-GUIDE	[ON] [OFF]
-INCREMENTS	[1]
-TABS	[OFF] [LANGUAGE] [OTHER]

All of these edit options for a file can be changed after the file is created using HPToolset's SET command except for SOURCE LANGUAGE, LINELENGTH, and FILESIZE.

You can also set parameters which change the environment in which HPToolset's editor runs. You can use the SET ENVIRONMENT command to set the following parameters at any time.

```
-ECHO =   [ON] [OFF]
-QUIET =  [OFF] [ON]
-N = integer
-EDITMODE [VISUAL] [COMMAND]
-FIND STRING = string
-PAGESIZE = integer
-CHANGE STRING = string
```

o Program Translation:

The translation phase of program development (compiling the source program into a USL file and preparing a program file from the USL file) no longer needs to be explicitly performed by going through the HP 3000 Compiler and Segmenter. To provide ease of use and efficiency, HPToolset uses screen-labeled function keys which enable you to COMPILE, PREP, and RUN COBOL II programs in a single step using the GO key, or in separate steps using the COMPILE, :PREP, and :RUN keys. The compilation is done in "Background", thereby allowing other concurrent tasks.

The latest online listings are saved when files owned by your Workspace are compiled. This allows you to see the listing on your terminal, rather than having to wait for a hard copy. Any compilation errors may be easily corrected. Soft keys defined by the HPToolset EDITOR will allow you to directly display and modify incorrect lines identified on the compiler listing, without the necessity of opening and scrolling through the entire source file.

When HPToolset creates a Workspace, a menu is displayed showing defaults for the USL and Program file names, as well as Prep options and the compile priority queue. A file name is the Workspace name with a U (for USL file) or P (for Program file) appended. Changes to these default values are accomplished by simply retyping over the existing text. These options can be changed at any time. By associating a USL and Program file with the Workspace, HPToolset frees you from having to explicitly specify these file names with COMPILE, :PREP, and :RUN.

Source files are named when you create the edit file or CONVERT an ASCII file into your Workspace. Since the $INCLUDE and COPY statements may cause the compiler to read source input from many files, source files may actually expand to a set of files within your Workspace. HPToolset will compile any source file, including a source file not in your Workspace, as long as it is in TSAM format and has been made known to your Workspace through the USE command.

- Symbolic Debug:

  COBOL II Symbolic Debugging significantly cuts testing and debugging times. The Symbolic Debug feature allows you to debug your COBOL II program within the HPToolset environment. In this way, you have complete interactive debugging capability while eliminating the tasks of converting octal numbers into meaningful data values, and code addresses into source statements.

  COBOL II application programmers who are apprehensive about using MPE debug will welcome the ease and swiftness with which COBOL II programs can be debugged using HPToolset.

  Symbolic Debug allows you to easily perform the following functions when monitoring the execution of your program:

  1. Set or clear breakpoints symbolically by (qualified) paragraph names, section names, or compiler generated source listing line numbers.

  2. Allow qualification of symbolic locations by main or subprogram name.

  3. Specify a list of HPToolset commands which will automatically execute at a specified breakpoint.

  4. Monitor a data item. Display a message or (optionally) interrupt program execution if the value changes (DATATRACE). Specify a list of HPToolset commands you want to automatically execute when the value of this data item changes.

  5. Easily display data items according to their PICTURE format or a specified number base (octal, integer, character, hexadecimal).

  6. Change the contents of a data item. Conversions are done by the rules for the COBOL MOVE statement which include alignment by decimal point, padding, and truncation.

  7. Display section and paragraph names as they are executed (TRACE) or display the last n paragraphs your program executed before the present location (RETRACE).

  8. Display the name of the MAIN program, currently "CALLed" subprograms, and the "CALLing" location (CALLS).

  9. Access MPE DEBUG from HPToolset.

Hardware Requirements

Recommended processors are the HP3000 Series III, 40, 44, or 64 with a minimum of one Megabyte of main memory.

A VPLUS/3000 compatible terminal is required to fully support all the screen functions of HPToolset. These terminals include the 2382A, 2622A, 2624A/B, 2626A, 2645A, 2647A, and the 2648A.

Software Requirements

The MPE IV operating system which has the C.00.20/D.00.20 version of MPE. Prerequisite is the A.00.08 version of the COBOL II compiler and the A.01.07 version of Segmenter.

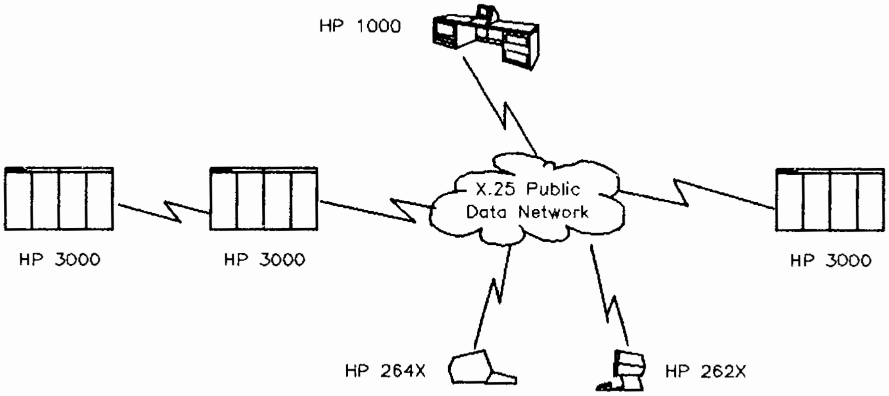
Documentation

HPToolset Reference Manual (32350-90001).

Introducing DSN/DS/X.25

by Carol Hibbard, Information Networks Division

The DSN/DS software has been enhanced to provide support for CCITT recommendation X.25, which allows for system-to-system communication across a packet-switching network. HP's implementation of X.25 protocol can also be used on leased or hardwired lines. Support has also been added for CCITT recommendations X.3/X.28/X.29, which allow asynchronous character mode terminals to communicate with remote systems via a packet switching network for the HP 3000 computer family (Series II/III/30/33/40/44/64). The following diagram illustrates these new DSN/DS capabilities:



These new capabilities offer a new cost alternative, increased reliability, and simplified interconnection of systems, especially for international connections. Packet-switching network charges are based on volume of data rather than distance. These networks provide redundant paths for recovery from internal failures and gateways conforming to recommendation X.75 for access to other X.25 public data networks.

The DS/X.25 software can provide better performance than the bisync version of DS on leased or hardwired lines. The X.25 protocol is full-duplex, so line throughput can be increased by a factor of two if there are users on both ends of the line; however, CPU utilization increases also. Refer to the HP 3000 Computer Systems Performance Guide (5953-7441) for additional information. Performance on network lines is highly unpredictable depending on the load at any given time.

The user interface for system-to-system communication is the same except for the DSLINE command, which now accepts a node name, LDEV number, or class name to specify the DS device. Node names are specified in the network database, which is manipulated by a new utility called NETCONF. Asynchronous character mode terminals connect to X.25 networks via a special facility provided by the network called a PAD (Packet Assembler/Disassembler). The network attempts to make the PAD's presence invisible, but there some noticeable differences between a terminal directly connected to the HP 3000 and one connected through a PAD. The DSN/DS HP 3000 To HP 3000 Reference Manual (32190-90001) has four new appendices containing information on DS/X.25. Appendix A contains the MPE SYSDUMP dialogue, Appendix B contains new error messages, Appendix H documents the NETCONF utility, and Appendix J contains information on using the new X.25/X.29 capabilities.

These new enhancements are included in version A.04.03 of DS, which requires at least version A.05.06 of CS and the 2208 release of MPE. Support for X.25 network connections is available only on the X.25 public data networks on which HP has been certified. Networks certified at this time are Telenet and Tymnet in the United States, Transpac in France, and Datapac in Canada. For status on certification with Public Data Networks not listed here, consult your local HP Sales representative.

Greater Graphics for DSG/3000

by Robert M. Studer, Information Networks Division

The A.01.00 release of Hewlett-Packard's DECISION SUPPORT GRAPHICS/3000 (HP DSG/3000) is highlighted by important improvements in three key areas:

o Product Features
o Peripheral Support
o Subsystem Support

Product Features

o DSG/3000 users now have the capability of specifying the size, color and font style for:

Titles - Main, Sub, Left Axis, Bottom Axis
Labels - Segment, X Axis, Y Axis
Legend Text
Footnote(s)

The size specification has been modified to correspond to printing industry standard point sizes (72 points/inch). This modification provides for consistency with other IND Office Systems software products (IFS/3000, IDS/3000, HPDRAW). Text may range in size from 1 to 72 points relative to a standard 8.5" x 11.0" paper size. For larger paper sizes, the text is scaled (up) accordingly. The color specification allows text to be generated in any one of up to 16 (depending on the device) colors. The font style specification provides for software text generation in any of 4 font styles; Simplex Roman (Stick), Simplex Script, Triplex Roman and Gothic English. The font style specification feature is also provided for textual annotations. Additionally, the capability to specify one of six native languages (Swedish/Finnish, Norwegian/Danish, French, German, United Kingdom, Spanish) for the Simplex Roman (stick) software and hardware (plotters only) character font is provided.

o The limitation of 31 bars on a single chart has been increased to 60. This provides for generation of a chart with 5 years of data by month or one year of data by weeks.

o The ability to generate a figure file (new file type) has also been implemented. This feature provides for the generation of a DSG/3000 chart in a form which may be used as input to Hewlett-Packard's TEXT AND DOCUMENT PROCESSOR (TDP/3000) in order to produce electronically merged text and graphics on the HP2680A Laser Printing System.

o DSG/3000 now takes advantage of the buffer (1K) available in the new HP-IB

8-pen plotters (9872C/T). Additionally, the polyfill routines utilize hardware types for increased speed. When DSG/3000 is used in conjunction with the HP2703A and HP2627A color graphics terminals, the hardware polyfill capabilities of these devices is employed.

o An additional plotting speed selection is now available on the GRAPHING OPTION menu. Selection of T, for "transparency", will cause a fifteen (15) minute pause between selection of pens when drawing bars or pie segments to prevent adjacent inks from bleeding into one another. The length of the pause is user-configurable via the DSG/3000 message catalog (SM1S250A.HP32250.SUPPORT).

Peripheral Support

o DSG/3000 maintains pace with the rapidly expanding peripheral product offering of Hewlett-Packard. The A.01.00 release of the product provides direct support of the following new peripherals:

HP2680A Laser Printing System
HP7470A Graphics Plotter
HP7585A Drafting Plotter
HP2382A Office Display Terminal
HP2703A Color Graphics Terminal
HP2627A Color Graphics Terminal

Subsystem Support

o DSG/3000 now provides support for three new Hewlett-Packard software products: HPDRAW, HPEASYCHART and Pascal, and an existing communications software product: MTS/3000.

o Programmatic users of DSG/3000 may now generate application programs in Pascal. Language code 5 is provided to indicate Pascal as the calling language to GINITGRAF.

o DSG/3000 provides support of HPDRAW, Hewlett-Packard's new, versatile presentation text and figure design system. This feature allows for the inclusion of a DSG/3000-produced chart in an HPDRAW-produced visual aid.

o DSG/3000 also provides support of HPEASYCHART, Hewlett-Packard's new, "no experience necessary" chartmaker. This feature allows for the retrieval and modification of HPEASYCHART-produced charts.

o The benefits of DSG/3000 are now available to those customers utilizing Hewlett-Packard's Multipoint Terminal Support (MTS/3000) to support multiple terminals over a single data communications line.

Conversion

- o Current DSG/3000 users will be required to convert their chart files created with previous versions (A.00.00, A.00.01, A.00.02) in order to use them with version A.01.00. This is the result of changes in the chart file format made in order to implement the expanded feature set. Conversion is a non-destructive process. That is, chart files created under previous versions remain intact when the new version of the chart file is created. A conversion program, GUPDATE.PUB.SYS, is provided with this release of DSG/3000 to perform the conversion automatically. A new conversion procedure (intrinsic) is also provided for those users who wish to convert their existing chart files programmatically. The procedure name is GCONVERTFILE. Its format and parameters are as follows:

```
                  IA     BA          BA
    GCONVERTFILE (graf,chartfile,convertfile)
                 i/o     in          in
```

  where:

  graf is the integer array containing global information used by DSG/3000. The first word returns the call status.

  chartfile is the byte array containing the name of the chart file to be converted.

  convertfile is the byte array containing the name of the new chart file.

  Chart files need only be converted once. This intrinsic, therefore, should not be included on a permanent basis in any application program.

- o Programmatic users of DSG/3000 will be required to, at the very minimum, rePREP their application program(s) with a larger (by 12K words) ;MAXDATA= value (i.e., new *maxdata* = old *maxdata* + 12000). This is the result of changes made in the low level graphics routines. This situation (i.e., rePREP only) will exist only if the user's program(s) currently have the GRAF communication area defined as 1600 words (3700 words total for BASIC) or larger in size. If this is not the case, then it will be necessary to change the GRAF communication area size to a minimum of 1600 words for SPL, FORTRAN, COBOL and Pascal (3700 words total for BASIC) and recompile, in addition to rePREPing with an additional 12K words for the ;MAXDATA= parameter.

- o Additional group capabilities are now required to successfully execute this version of DSG/3000. At the very least, the DS ('Extra Data Segments') capability-class attribute is required. The specification of the HP2680A Laser Printer as a direct output device requires the additional capability-class attributes of PH and MR ('Process Handling' and 'Multiple-RIN'). Programmatic users will be required to rePREP their application programs with these capabilities.

SUMMARY

These enhancements to DSG/3000 provide its users even greater flexibility and control than previously, reflecting Hewlett-Packard's commitment to business graphics and the concept of Integrated Information Management. For complete details and/or a demonstration, contact your local Hewlett-Packard sales office.

Introducing HPDRAW

by Ellen Brigham, Information Networks Division

HPDRAW is an easy to use business presentation-aids design program. It can combine text, geometric primitives, charts from DSG/3000 or HPEASYCHART, or figures created by the user or taken from a library file. HPDRAW drawings can be plotted on transparency film or paper, or may be sent directly to the HP 2680 Laser Printer. Drawings created using HPDRAW may be incorporated in documents produced with TDP/3000 or HPWORD; the merged text and graphics are printed on the HP 2680 Laser Printer.

HPDRAW was designed with the end user in mind. Menus are presented, from which the user can select various tasks using the terminal's function keys. HPDRAW can be used by clerical or professional persons to create quality presentation aids in a short time.

HPDRAW Features

o Flexible, friendly, and menu-driven.

o Four high quality fonts in a wide range of sizes.

o Multi-color output when plotted on an HP multi-pen graphics plotter.

o Lines, arcs, circles, and boxes can be easily created.

o May utilize charts created with HPEASYCHART or DSG/3000.

o Easy revision of drawings.

o Wide variety of supported peripherals.

o Drawings can be included in TDP/3000 or HPWORD documents.

Hardware Environment

The minimum recommended hardware system is an HP 3000 Series III, 30, 33, 40, 44, or 64, with one megabyte of memory. Actual memory requirements will vary, depending on overall system load and response time needs.

o Supported terminals:

HP 2623A
HP 2647A/F
HP 2703A
HP 2627A
HP 2648A

- o Supported HP-IB plotters - the following plotters are supported when configured as a shared peripheral with an HP 2647A/F or HP 2648A graphics terminal:

  7225A/B (with personality module 17601A)
  7245A/B
  7470A (option 002)
  7580A (option 002)
  7585A (option 002)
  9872A/B/C/S/T

- o Supported RS-232 plotters - the following are supported in "eavesdrop" mode between terminal and computer, or when connected to a separate port:

  7220A/C/S/T
  7221A/B/C/S/T
  7225A/B (with personality modules 17603A and 17604A)
  7240A
  7470A (option 001; option 016 for eavesdrop)
  7580A (option 001; no "eavesdrop")
  7585A (option 001; no "eavesdrop")

- o Supported printer - HPDRAW supports the HP 2680A Laser Printer (requires the C/D.01.00 release of MPE, and the printer upgrade package).

Software Environment

HPDRAW requires the MPE IV Operating System (release 2134 or later), and VPLUS/3000 (which is part of the Fundamental Operating Software).

HPDRAW may be run via data communications links operating under Multipoint Terminal Software (MTS) or DSN/DS.

Training and Ordering Information

HPDRAW has a Reference Guide, which is included with the product, and a self-paced training package, which is purchased separately.

Contact your local HP Sales and Service Office for further information.

HPEASYCHART: Quick and Simple

by Kim Sasko, Information Networks Division

HPEASYCHART is HP's all new "no-experience necessary" chartmaker designed for use by secretaries, business professionals, and managers. An inexperienced user can produce line charts, bar charts, pie charts, and scattergrams interactively within 5 to 10 minutes. These charts can be displayed on an HP graphics terminal or drawn on either paper or overhead transparencies using an HP multipen plotter.

The "follow-by-example" selection menu assists novice users in selecting a chart by its visual appearance on an HP graphics terminal, rather than through a written description. This visual approach to chart selection and the simplicity of the user interface make HPEASYCHART an ideal graphics product for users unfamiliar with computer programming, and for the more experienced users who wish to quickly create a chart.

HPEASYCHART Features

o Picture-Oriented Main Menu - users may select a chart by its appearance.

o Scrolling Data Entry - allows up to 75 data points for five variables to be entered on the terminal screen and then plotted in a chart.

o Multiple Charts on a Page - positions charts on the top, bottom, left, or right halves of a page, as well as in any quadrant, for a total of up to four charts per page.

o Automatic Scaling - scales charts automatically for different paper sizes by selecting a predefined size (i.e. 8.5 X 11 and 11 X 17, European A3 and A4, D size flip charts).

o Color and Texture Selection - provides up to eight colors and eight textures for lines and areas.

o Wide Variety of Supported Devices - previews charts on any HP graphics terminal before plotting. For chart-definition only, non-graphics terminals which support VPLUS/3000 may be used to fill in the menus and direct the finished chart to a plotter.

o Built-in HELP Facility and Examples - speeds up the production process by eliminating the need to design a chart from beginning to end.

o Charts are Upward-compatable with Decision Support Graphics/3000 - allows users to add text fonts, annotations and figures in order to enhance an HPEASYCHART chart.

- o High-Quality Visual Aid Production - charts may be included in drawings created by HPDRAW, allowing the user to create a first-class aid on paper or transparency.

Hardware Environment

HPEASYCHART runs on hardware which supports DSG/3000. The minimum hardware system needed to implement DSG/3000 is an HP 3000 Series III, 30, 33, 40, 44, 64 with 1Mb of memory.* However, the recommended amount of memory varies dependending on the expected overall system load, and performance and terminal response time requirements.

o Supported Terminals:

For on-screen display and chart definiton	For chart definition only	
HP 2623A	HP 2382A	HP 2641A
HP 2647A/F	HP 2624A/B	HP 2642A
HP 2648A	HP 2626A	HP 2645A
HP 2703	HP 2622A	
HP 2627A		

o Supported HP-IB Plotters - the following HP-IB plotters are supported when configured as a shared peripheral with a 2647A/F or 2648A graphics terminal:

Vector Pen Plotters	Thermal Plotter/Printer
9872A/B/C/S/T	7245A
7225A/B with 17601A Personality Module	
7470A Option 002	
7580A Option 002	
7585A Option 002	

o RS-232 Plotters - the following RS-232 plotters are supported by HPEASYCHART when configured in "eavesdrop" mode between terminal and computer, or when directly connected to a terminal port:

Vector Pen Plotters	Thermal Plotter/Printer
7220A/C/S/T	7240A
7221A/B/C/S/T	
7225A/B with 17603A (no eavesdrop) or 17604A Personality Modules	
7470A Option 001/016	
7580A Option 001 (no eavesdrop)	
7585 Option 001 (no eavesdrop)	

*Terminal baud rates greater than 2400 are supported only on Series 40, 44, and 64.

o Supported Printers - HP2680A Laser Printer (requires C.00.20/D.00.20 version of MPE and the laser printer upgrade package)

Software Environment

HPEASYCHART requires the MPE-IV Operating System (C.00.02 version of MPE or later), and VPLUS/3000, which is part of the Fundamental Operating Software (FOS).

HPEASYCHART may be run via data communications links operating under Multipoint Terminal Software (MTS) or DSN/DS.

Training and Ordering Information

Because of the simplicity of HPEASYCHART, there is no customer training available. However, a tutorial section is included in the HPEASYCHART reference manual.

Please contact your local field sales office for a demonstration of HPEASYCHART and for additional information on the ordering of documentation, materials and software support services.

SECTION II:
GENERAL INFORMATION

Using Labeled Tapes

by Harvey Skinner and Bob Thomas, Computer Systems Division

Editor's note: This article was originally published by Mike Stodghill in Communicator Number 27. It is being re-published here with modifications and additions.

Labeled tapes: Basic concepts

Labeled tapes are intended to provide:

o A permanent identification for tape reels, or volumes.

o Files which extend over more than one volume.

o More than one file on a volume.

o Retrieval of files by file name.

o Additional security, to protect against unauthorized erasure or access to files.

When a tape volume is first written, the operator should assign to it a unique identifier of up to six alphanumeric characters. This identifier is the Volume Name. It is often strictly numeric, and volumes in an installation's library can be stored and sorted by this number. A collection of volumes containing one file or a related group of files is called a volume set.

Facilities used with MPE Tape Labels include:

o Commands
  :FILE COMMAND
  :SHOWDEV COMMAND

o Intrinsics
  FOPEN
  FREAD
  FWRITE
  FCONTROL
  FWRITELABEL
  FREADLABEL
  FCHECK

o New Console Messages

A working knowledge of these features is assumed. In the following discussion only those portions of the user interface which are relevant to MPE tape labels will be explained. For further information please see:

- MPE Commands Reference Manual (30000-90009).

- MPE Intrinsics Reference Manual (30000-90010).

- MPE Console Operator's Guide (32002-90004).

- MPE File System Reference Manual (30000-90236).

Volume Set

A group of related volumes is called a volume set. The volume set ID is the volume ID of the first volume of the set; this will be discussed in more detail in the "Tape Label Format" section of this article.

When a user's data file spans more than one tape volume, MPE will insure that the appropriate volume of the set is mounted by issuing a mount message to the console; the user's program will be suspended until the operator has completed mounting the appropriate volume. Once the volume is mounted, the Automatic Volume Recognition function will wake the user program and processing will continue.

Commands Used With Labeled Tapes

o :FILE COMMAND

Users must specify optional file label information in the :FILE Command. Partial syntax for the FILE command is:

```
:FILE formaldesignator=filename

   [;NOLABEL                                    ]
   [;LABEL [=[volid],[type],[expir_date],[seq]]]
```

where

filename — Is up to eight alphanumeric characters, beginning with an alphabetic character, that names the file to be processed. This may be further qualified with a lockword and group; since there is no account associated with the file on tape, no account name is specified.

If the lockword is used, certain other conditions must be considered. If byte 54 of the HDR1 label is a space, there is no security protection for the file. In any case, MPE tape labels will ignore security for

IBM Standard labels. Security will also be ignored for ANSI Standard labels unless byte 54 = %23. If a lockword is specified on output then byte 54 of HDR1 label will be coded %23 and the lockword will be written in the portion of HDR2 label reserved for the operating system. This will enable HP 3000 users to read ANSI Standard and IBM Standard tapes written by other systems, but will provide lockword security for ANSI Standard labeled tapes written by MPE on the HP 3000.

NOLABEL — Specifies that this is not a labeled tape, and turns off FOPTION bit 6 in FOPEN.

LABEL — Specifies that this is a labeled tape, and is equivalent to FOPTION bit 6 in FOPEN.

volid — Up to six printing characters that identify the volume. (Optional when opened for output.)

type — Three characters that specify label type. (Optional parameter; default is FOPEN type.) Options are:

ANS - ANSI standard labels
IBM - IBM standard labels

expir_date — In the format MM/DD/YY. This is the expiration date of the file, or the date after which information contained in the file is no longer useful. The files can be overwritten after this date. (Optional parameter, with default 00/00/00 meaning the file can be overwritten immediately.)

seq — Up to four characters that denote the position of the file in relation to other files on the tape. "0" will cause a search of all volumes until file name is found. If *seq* is "ADDF" then the tape will be positioned to add a new file on the end of the volume (or last volume in a multi-volume set). If *seq* is "NEXT" then the tape will be positioned at next file on the tape. If this is the first open, then "NEXT" will position the tape at the first file./ If a rewind took place on the previous CLOSE, then the tape will remain at the beginning of the previous file. (Optional parameter; default is FOPEN specified.)

o :SHOWDEV COMMAND

Examples:

```
:SHOWDEV TAPE

LDEV    AVAIL         OWNERSHIP           VOLID

  7     AVAIL
  8    UNAVAIL      #S81: 1 FILES         SADSAM           (ANSI)
  9     AVAIL
 12     AVAIL                                              (NOLABEL)
 13     AVAIL                             TAPE01           (ANSI)
 14     AVAIL                             666666           (IBM)
 15     AVAIL

:SHOWDEV 8

LDEV    AVAIL         OWNERSHIP           VOLID

  8    UNAVAIL     #S81: 1 FILES          SADSAM            (ANSI)

:SHOWDEV 7

LDEV    AVAIL         OWNERSHIP           VOLID

  7     AVAIL
```

In the examples above, logical devices 7, 8, 9, 12, 13, 14 and 15 are tape drives. Volumes are mounted on 8, 12, 13, and 14, but only LDEV 8 has an active volume. No volume is mounted on 7, 9, or 15. The volume on LDEV 12 is unlabeled. (Note that, in these examples, the DEN and ASSOCIATION fields have been omitted from the command output.)

Intrinsics

o FOPEN

The FOPEN intrinsic can also be used to specify label information. The format is:

FOPEN (*formaldesignator*, *foptions*, *aoptions*, *recsize*, *device*, *formsmessage*,
userlabels, *blockfactor*, *numbuffers*, *filesize*, *numextents*, *initialloc*,
filecode)

Users desiring to specify labeled tape information in the FOPEN intrinsic must set *foptions* bit 6 to "1".

Additionally, tape label information must be placed in the forms-message parameter of the FOPEN intrinsic. (Note: Limit of 49 characters does not apply to labeled tapes.) The tape label information must be formatted as follows; parameters are the same as those described in ":FILE COMMAND" above:

.[*volid*],[*type*],[*expir_date*],[*seq*];

When opening files on a labeled tape:

1. You must specify tape label information with a :FILE command or the *formsmsg* parameter of the FOPEN intrinsic.

2. There must be an FOPEN/FCLOSE sequence for each file on a volume set. Simultaneous opens of files on labeled tapes are not allowed.

Example:

To open file named INPUT on tape labeled TAPE01, use

```
    :FILE INDATA=INPUT;DEV=TAPE;LABEL=TAPE01,ANS,,0
```

or for FOPEN, use:

```
    MOVE LABELINFO:=".TAPE01,ANS,,0;";

    FILENUM:=FOPEN(FILID,%1005,5,,DEV,LABELINFO);
```

o FREAD

The FREAD intrinsic is used to read files. The CCG condition code has been changed for labeled tapes. CCG will be returned at actual EOF. CCG with a transmission log of 0 will be returned on attempts to read past EOF. CCG will not be returned for EOT where an actual file spills over onto another volume. After CCG on EOF, only FREADLABEL or FCLOSE will be accepted. CCL and CCE are unchanged.

o FWRITE

The FWRITE intrinsic is used to write files. When writing files on a scratch volume, or when subsequent volumes of a multivolume file are required, it is not necessary to specify volume identification in the :FILE command or the FOPEN intrinsic. MPE will request this information at the operator's console if it is not supplied.

Once the volume has been positioned by the FOPEN intrinsic, The expiration date of the file will be checked. If the file has expired, the existing header label will be overwritten with a header label for the file to be written. If the file has not expired, MPE will ask the operator if the file may be overwritten.

It is recommended that only the last file on a volume be overwritten, since files following the overwritten file will not be accessable at a later time. Files can be appended to existing volumes by specifying ADDF in the *seq* parameter of the :FILE command or the FOPEN intrinsic; this will cause the tape to be positioned to just past the last file on the volume.

No CCL will be returned at EOT. If an FWRITE encounters EOT, MPE will write EOV1 AND EOV2 labels and initiate reel switching. Otherwise the condition codes are unchanged.

Example of writing a labeled tape:

Use :FILE command:

```
:FILE NEWTAPE1;DEV=TAPE;LABEL=FIL099,ANS,12/31/86,NEXT
```

or *formsmsg* parameter of FOPEN intrinsic:

```
BYTE ARRAY LABELID (0:79):=".FIL099,ANS,13/31/86,NEXT";

FN02:=FOPEN(FILID2,%1004,5,,DEV,LABELID);
```

- FCLOSE

In order to maintain position when creating or reading a volume set, the disposition at FCLOSE should be either 2 (rewind to beginning of current file) or 3 (no rewind position at next file). A disposition of 0 or 1 (rewind and unload) implies close of an entire volume set.

For example, a user desiring to close the current labeled file and open the same file again must close the file with a disposition code of 2. The file will be rewound to its beginning but will not be unloaded. Subsequent requests to open the file will not reposition the tape, if *seq* = NEXT or default.

- FWRITELABEL/FREADLABEL

User labels contain information specified by the programmer that is related to a particular file. The intrinsics FREADLABEL and FWRITELABEL are used to create and read both header and trailer user labels.

In order to write a user header label, the programmer must issue the FWRITELABEL intrinsic prior to the execution of the first FWRITE intrinsic. After issuing the FWRITE intrinsic, MPE will terminate the data area and write user trailer labels for any FWRITELABEL request.

To read user header labels, the user must issue the FREADLABEL intrinsic prior to the execution of the first FREAD. The execution of the first FREAD causes MPE to skip past any unread user header labels.

User trailer labels must be written after the last FRWITE is issued to the file and before closing the file. After an FWRITELABEL is issued, additional

FWRITEs to the file will not be accepted; an error will be returned if an attempt is made. An attempt to read more user header or user trailer labels than actually exist will result in a CCG.

User trailer labels can be read by issuing an FREADLABEL after any of the data areas of the file is read or processed. MPE assumes that the next action to be taken after reading user trailer labels will be to close or rewind the file.

NOTE

> All ANSI Standard and IBM Standard labels are 80 bytes in length. User header labels and user trailer labels should conform to this standard. Any attempt to write or read other than 80 bytes (or 40 words) will result in CCL.

- FCONTROL

when processing labeled tapes. This is because labeled tapes are file oriented, with the possibility of multiple files on a single tape, rather than device or single file oriented as is the case with unlabeled tapes. Thus FCONTROL operations such as "Space forward to tape mark" cannot be allowed when using labeled tapes.

Certain other FCONTROL functions might yield unexpected results if the user issued them. For instance, the "Rewind file" control will position the tape to the beginning of the opened file, not to the beginning of the volume. A list of the FCONTROL control codes and their new or modified meanings follows:

Control Code	Meaning	Operating with Labeled Tapes
5	Rewind File	Position tape at the beginning of the opened file on the volume. (Start of user Header Labels, if any.)
6	Write End-Of-File	Not allowed for labeled tapes.
7	Space Forward to Tape Mark	Space forward to start of User Trailer Labels, if any.
8	Space Backward to Tape Mark	Rewind file to start of User Header Labels, if any.
9	Rewind and Unload	Not allowed for labeled tapes.

In the case of a multi-volume configuration, when the user requests rewinding of the file that spans the volume, MPE will not request that the previous volume be mounted. It will only rewind the file to the beginning of the current volume.

- o FCHECK

Errors associated with labeled tapes:

116	INVALID TAPE LABEL FOPEN PARAMETERS
117	ATTEMPT TO WRITE ON UNEXPIRED TAPE FILE
118	INVALID HEADER OR TRAILER TAPE LABEL
119	I/O ERROR POSITIONING TAPE FOR TAPE LABELS
120	ATTEMPT TO WRITE IBM-STANDARD TAPE LABEL
121	TAPE LABEL LOCKWORD VIOLATION
122	TAPE LABEL TABLE OVERFLOW
123	END OF TAPE VOLUME SET
124	ATTEMPT TO APPEND LABELED TAPE

Reel Switching

When processing a multivolume set, the user should specify the first volume to be processed in the :FILE command or the FOPEN intrinsic. If this is omitted, as might be the case for a scratch tape, the operator is prompted for the information. When an end-of-tape reflective marker is encountered during a write, MPE will request that the next volume be mounted on the same LDEV. A labeled tape mounted to be rewritten will be checked for expiration.

Tape Security

MPE provides two kinds of tape file protection. Protection against accidental destruction is an integral part of MPE, because the user must request specific volume and file information before a file can be overwritten. Protection of private information is provided by the user with the *lockword* parameter of the :FILE command and FOPEN intrinsic. When a file is created with a lockword specified, all subsequent users of the file must provide that lockword.

Overwriting Tape Labels

A tape label can be overwritten with other information using FCOPY. For example, suppose we want to "erase" the Volume Header and the HDR1, HDR2 labels. As these are the first three records on the volume, we can create a disc file BLNKFILE consisting of blank records and copy it to the device file representing the mag tape. Thus the following command sequence will copy our three blank records to the tape volume:

```
:FILE FTAPE;DEV=TAPE
:FCOPY FROM=BLNKFILE; TO=*FTAPE
```

If this tape volume is now rewound and reloaded, it will be recognized by MPE as an unlabeled tape. Label scratching should not be done if the user wants to maintain the integrity of the data records on the tape. Generally speaking, mag tape is not designed to enable the altering of only a few specific records without jeopardizing the integrity of other information on the tape. Refer to the MPE Intrinsics Reference Manual (30000-90010) for more information on altering individual records on tape.

ANSI and IBM Standard Labeled Tapes

There are two labeled tape standards that are HP supported: ANSI and IBM. The main differences in the two are that IBM labels are written in EBCDIC, and ANSI labels are written in ASCII. IBM labels also differ in that their second file header is in a slightly different format. The MPE tape label facility can read and write labeled tapes that conform to the ANSI standard, but IBM labels can only be read. Only ANSI standard tapes support lockwords.

When To Use Labeled Tapes

Labeled tapes lend themselves well to an application where a file or group of files spans more than one volume, as the labels contain the necessary title and sequencing information. Labeled tapes also provide an extra level of security as the operator is informed of the volume name and the volume set name when the tape is mounted, thus reducing the chance of inadvertently using the wrong tape. The labels also contain other useful information such as creation date and expiration date, which otherwise may not be saved with the file or volume. Labeled tapes provide an extra level of security with MPE STORE tapes in saving and backing-up user files.

Labeled tapes are very convenient if a user is streaming jobs and doesn't know when a particular tape will be accessed. In this case, the tape can be mounted before it is required by a job, and MPE will then access the tape when needed without further operator intervention.

There are applications that do not lend themselves to labeled tapes. A cold load, for example, requires a bootstrap and other information to be present on the first records of a tape; labeled tapes contain label information there. While it is possible to do a SYSDUMP to a labeled tape, its use would be severely limited since the user would not be able to boot from it.

Console Messages

One of the following messages is displayed when a tape is recognized. In the case of ANSI tapes, message (3) will be displayed if the *volid* and the *volsetid* are the same; otherwise, message (4) will be displayed.

(1) *hr:min/pin*/Vol(unlabelled) mounted on LDEV#*n*

(2) *hr:min/pin*/Vol(IBM) mounted on LDEV#*n*

(3) *hr:min/pin/*Vol(ANS) mounted on LDEV#*n*

(4) *hr:min/pin/*Vol *volid* (ANS) of *volsetid* mounted on LDEV# *n*

When a labeled tape is opened but has not been mounted, message (5) is displayed. A reply of "0" will abort the FOPEN; otherwise the operator should mount the requested volume without any reply, or mount a blank tape and reply with the LDEV#.

(5) *hr:min/#jsnum/pin/*Mount tape of volume set *volid* (ANS)

When a file spans more than one volume and it is necessary to switch volumes, message (6) or (7) is displayed. The operator should mount the appropriate volume on the specified device. No reply is necessary.

(6) *hr:min/#jsnum/pin/*Mount next volume of set *volid* on LDEV# *n*

(7) *hr:min/#jsnum/pin/*Mount prior volume of set *volid* on LDEV# *n*

When an unlabeled tape is mounted in response to message (6), message (8) is displayed. The operator is expected to supply a Volume Identification which will be written on the tape by MPE.

(8) *hr:min/#jsnum/pin/*Vol ID for volume of set *volsetid* on LDEV#*n*?

When a labeled tape is opened and no volume header information is supplied, message (9) is displayed. The operator is expected to reply with Volume Identification.

(9) *hr:min/#jsnum/pin/*Volume ID for *filename* (MAX CHAR.=6)?

Message (10) is displayed when a labeled, unexpired tape is opened as unlabeled, or when a labeled, unexpired tape is mounted in response to message (5) or (6).

(10) *hr:min/#jsnum/pin/*OK to write on unexpired vol (*volid*) on LDEV#*n*? (Y/N)

Message (11) is displayed when a labeled, expired tape is opened as unlabeled, or when a labeled, expired tape is mounted in response to message (5) or (6).

(11) *hr:min/#jsnum/pin/*OK to write on expired vol (*volid*) on LDEV#*n*? (Y/N)

Tape Label Format

In accordance with the ANSI standard, a labeled tape contains a series of label records at the start of each volume, before and after each file, and at the end of a tape volume when a file set is to be continued on another volume. The format of each label written by MPE is described in the following paragraphs. Each label shown below spans 1 record (80 bytes). The user can access label information with the FFILEINFO intrinsic, as described in the MPE Intrinsics Reference Manual (30000-90010).

The first label on a volume is the Volume Header Label, whose format is shown below (CP indicates character position):

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"VOL"
4	Label Number	1	"1"
5-10	Volume Identifier	6	Character string to identify the volume
11	Accessibility	1	"0" if IBM, else " "
12-79	NOT USED	62	BLANKS
80	Label-Standard Version	1	"1" if HP ANSI, else " "

Note that all of the information in the Volume Header Label is supplied by MPE, with the exception of the Volume Identifier, which is supplied as either the *volid* in the LABEL parameter of the :FILE command, or in the *formsmsg* parameter of the FOPEN intrinsic. The volume identifier (which is sometimes called the Volume Name) is intended to uniquely identify a particular tape reel. For this reason, a given Volume ID should not be used more than once at an installation.

The Volume Header Label is followed by two File Header Labels. These header labels appear at the start of each volume, and before each file in a file set. The format of the first File Header Label is shown below:

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label ID	3	"HDR"
4	Label Number	1	"1"
5-21	File Name	17	File Name
22-27	Volume Set ID	6	Name of volume on which this file set begins.
28-31	Reel Number	4	Counts the sequence of reels in the current file.
32-35	File Sequence #	4	Counts the sequence of files in this set.
36-41	NOT USED	6	BLANKS
42-47	Creation Date	6	yyddd; date of file creation. Preceded by one BLANK.
48-53	Expiration Date	6	yyddd; date on which file may be overwritten Preceded by one Blank.
54	Accessibility	1	%230 if lockword, "0" if IBM, else blank
55-60	Initial Block count	6	"000000"
61-73	System Code	13	"HP MPE 3000"
74-80	NOT USED	7	blanks

Several points should be noted regarding the first File Header Label. First, the File Name should usually not be duplicated within in a file set. Second, the Volume Set ID will be the same for all files in a set. Third, MPE will initialize and maintain all information in the HDR1 label except for the File Name and the Expiration Date. The File Name is supplied by the file creator through the :FILE command as the *formaldesignator* parameter at file creation, or as the actual designator after file assignment through a file equation. The Expiration Date is supplied by the user either through the LABEL parameter in the :FILE command or through the FORMSMSG parameter of the FOPEN intrinsic.

The Volume Set ID is simply the Volume ID of the first volume of the volume set. So, for the first volume, the Volume ID and the Volume Set ID will be the same. For the succeeding volumes in the set, the Volume Set ID shown in their header labels will be the Volume ID of the first volume. Note that MPE will always assemble the Volume Set ID in all HDR1 labels.

Following the first file header is a second file header record. Its format is:

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"HDR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed Length "V" = Variable length "U" = Undefined else undefined
6-10	Block Length	5	Block length in bytes
11-15	Record Length	5	Record Length in bytes
16-23	Lockword	8	MPE file lockword
24-36	NOT USED	13	MPE writes blanks.
37	Record Type	1	"A" = ASCII, "B"=BINARY
38	Carriage Control	1	"C"=control; " "=none
39/80	NOT USED	42	BLANKS

The information for the Record Format, Block Length, Record Length, and Record Type fields is provided through the REC parameter of the :FILE command, or through parameters in FOPEN. The file Lockword and Carriage Control information are also provided through :FILE or FOPEN. Label ID and Label Number are initialized and maintained automatically by MPE.

An optional label that may be added by the user is the User Header Label or UHL. A UHL may contain any information other than data. For example, a user may want to keep a generation code that is updated with each new version of a file. The UHL is written through the FWRITELABEL intrinsic.

Other than reads or writes by the user through FWRITELABEL or FREADLABEL, the UHL is ignored by MPE. (Tapes from non-HP facilities may contain User Volume Labels (UVLs); these are ignored by MPE.)

The format of a User Header Label is:

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"UHL"
4	Label Number	1	Label Number
5-80	Reserved for User application	76	Information of user's choice

MPE also writes trailer labels to file sets and/or volume sets on magnetic tape. If a file spans more than one volume, an End Of Volume (EOV) trailer label is written to indicate that the file is continued on another volume. Two EOV labels are written to consecutive records. The format of each is:

CP	Field Name	Length	Content
1-3	Label Identifier	3	"EOV"
4	Label Number	1	"1"
5-54	(SAME AS HDR1)	50	(SAME AS HDR1)
55-60	Block Count	6	Number of data blocks since the preceding HDR label set (Excluding tape marks and labels)
61-80	(SAME AS HDR1)	20	(SAME AS HDR1)

The second EOV label immediately follows EOV1; its format is similar to HDR2, as shown below:

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"EOV"
4	Label Number	2	"2"
5-80	(SAME AS HDR2)	76	(SAME AS HDR2)

Two trailer labels are also written by MPE at the end of each file: EOF1 and EOF2. Like the EOV labels, the EOF labels occupy two consecutive records on tape. The format of the First End Of File label is:

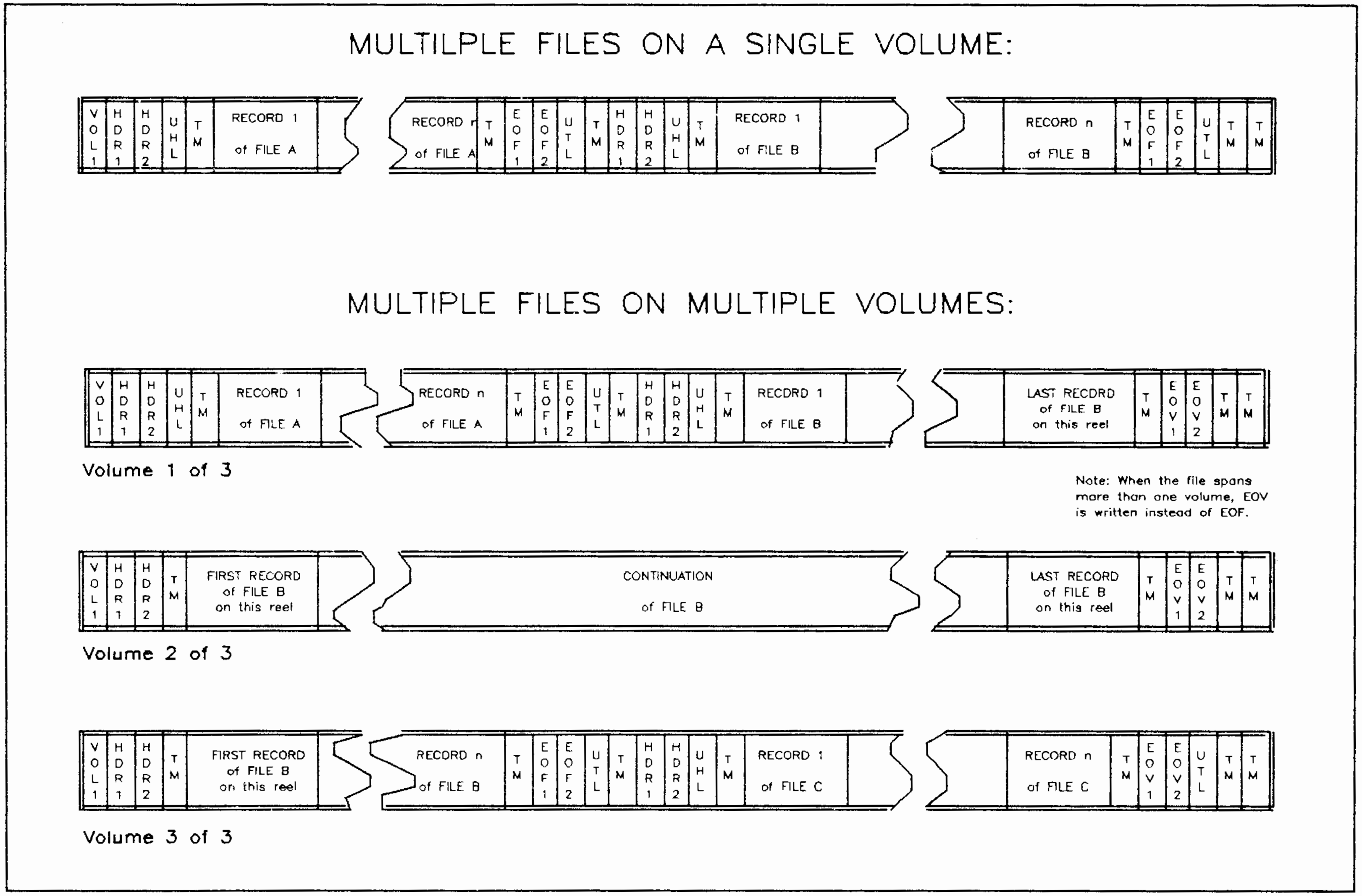
CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"EOF"
4	Label Number	1	"1"
5-54	(SAME AS HDR1)	50	(SAME AS HDR1)
55-60	Block Count	6	Number of data blocks since beginning of file label group
61-80	(SAME AS HDR1)	20	(SAME AS HDR1)

The format of EOF2 is similar to that of HDR2, as shown below:

CP	FIELD NAME	LENGTH	CONTENT
1-3	Label Identifier	3	"EOF"
4	Label Number	1	"2"
5-80	(SAME AS HDR 2)	76	(SAME AS HDR2)

Note that after the HDR1 and HDR2 labels are written, all information needed for the trailer labels is known to MPE. Thus the system writes all trailer labels (EOF and EOV) without the user again specifying this label information.

The following figure, which is from Appendix D of the MPE Intrinsics Reference Manual (30000-90010), illustrates how the various labels are laid out in a volume set:



MPE Tape Labels (Conforming to ANSI–Standard)

All About File Codes

by Ken Jordan and Greg Grimm, Computer Systems Division

There have been many inquiries about the nature and applications of file codes. We present the most common here:

o Why file codes?

MPE and applications programs use file codes as a way of identifying a type of file, thus insuring that the internals of the file are of the proper form. A good example is a program file. The MPE Loader opens a file and checks its file code. If it is equal to 1029, then the Loader will try to load the file; if it is not 1029, the file is not a legal program file and the Loader will generate an error message.

o How do I assign a file code to a file?

This can only be done when a new file is created using FOPEN or :BUILD. The file code is specified in the last parameter of the intrinsic FOPEN (*filecode*). Any user may specify a positive (NON-PRIV) file code, and a process running in privileged (PRIV) mode may specify a negative file code. If no file code is specified, 0 is the default. The file code becomes a permanent attribute of the file and cannot be changed. When building a file using the :BUILD of the Command Interpreter (CI), the file code may be specified by ;CODE=*filecode*. The code may be defined by a mnemonic or by an integer greater than or equal to 0.

o How can I find out the file code of a file?

You may use :LISTF,1 or :LISTF,2 on a file. The column marked CODE is really the file code. For some file codes, the CI has assigned a mnemonic. This mnemonic gives us a way to quickly decide what type of file we have. For example, a file code of 1024 has the mnemonic of USL, which tells us that we have a USL file. In addition to the :LISTF command, one could use LISTDIR2 to find out the file code. Programmatic access to the file code is done through FGETINFO (parameter 9) or FFILEINFO item 8.

o Does FOPEN check to see if the file codes match when I open an old file?

It depends on a) the type of file code requested, and b) the file code of the file. The following chart shows when FOPEN will require that the requested file code match the actual file code:

		REQUESTED FILE CODE	
		>=0	< 0
ACTUAL FILE CODE	>=0	FOPEN succeeds (Even if the file codes don't match)	FOPEN fails
	< 0	FOPEN fails	FOPEN succeeds if user runs in PRIV and filecodes match

As can be seen above, to gain access to a file with a negative file code, the user must be running in PRIV mode.

o What happens to file codes on FOPEN of a new file?

If you specify a positive file code that code is written into the the file label. If the file code is omitted, 0 will be written. If the user that called FOPEN with a negative file code is running in PRIV mode, the negative file code will be written; otherwise the FOPEN will fail.

o Can I :PURGE, :RENAME, :ALTSEC, :SETCATALOG, :STREAM a file with a PRIV file code (negative)?

You cannot do any MPE command to a file that contains a negative file code with the exception of :STORE and :RESTORE. The reason is that the command will try to open the file, and since it does not know the file code to expect, it will not request the proper file code. In addition, the CI does not allow the user to specify a negative file code in a file equation.

- Can I :STORE OR :RESTORE files with PRIV file codes?

  Yes, Store and Restore use ATTACHIO calls to find out the proper file code to use, and thus do not use the file system to access the file.

- If I can't use MPE commands (other than :STORE, :RESTORE) on a file with a PRIV file code, how do I perform functions usually done by the commands?

  The subsystem or user application that uses negative file codes must provide utilities to do basic MPE functions. An example of this is IMAGE's DBUTIL.

- Why would I want a negative file code?

  If a set of files have an interdependence, it would be desirable to prevent the files from being renamed, purged, etc. independently of each other. Since MPE commands do not work with negative-coded files, the application needs a utility which would be aware of the interdependence of the set of files.

- What effect do filecodes have on file system intrinsics other than FOPEN?

  After a PRIV file has been opened, accessing it also requires running in PRIV mode. This is to add security to the idea of PRIV file codes, so that only system code or supplied utilities can issue direct file system intrinsics to the file.

- Who can reserve file codes?

  Reserved file codes are assigned during operating system development. Thus, only specific Hewlett-Packard software products may have file codes reserved for them. File codes cannot be reserved by the customer.

- What are the reserved positive file codes and their mnemonics?

  They are:

Code	Mnemonic	Use
1024	USL	User Segmented Library
1025	BASD	Basic Data
1026	BASP	Basic Program
1027	BASFP	Basic Fast Program
1028	RL	Relocatable library
1029	PROG	Program File
1031	SL	Segmented Library
1035	VFORM	View Form file
1036	VFAST	View Fast Forms file
1037	VREF	View Reformat file
1040	XLSAV	Cross Loader ASCII file (SAVE)
1041	XLBIN	Cross Loader Relocated binary file
1042	XLDSP	Cross Loader ASCII file (DISPLAY)
1050	EDITQ	Edit quick file
1051	EDTCQ	Edit KEEPQ file (COBOL)

(Cont.)

Code	Mnemonic	Use
1052	EDTCT	Edit TEXT file (COBOL)
1054	TDPDT	TDP Diary file
1055	TDPQM	TDP Proof Marked QMARKED
1056	TDPP	TDP Proof Marked non COBOL file
1057	TDPCP	TDP Proof Marked COBOL file
1058	TDPQ	TDP Workfile
1059	TDPXQ	TDP Workfile (COBOL)
1060	RJEPN	RJE Punch file
1070	QPROC	QUERY Procedure file
1080	KSAMK	KSAM Key file
1083	GRAPH	GRAPH Specification file
1084	SD	Self describing file
1090	LOG	User Logging Logfile
1100	WDOC	HPWORD Document
1101	WDICT	HPWORD Hyphenation dictionary
1102	WCONF	HPWORD Configuration file
1103	W2601	HP 2601 Environment File
1110	FCELL	IFS/3000 Character Cell file
1111	PFORM	IFS/3000 Form file
1112	P2680	IFS/3000 Environment file
1113	PCCMP	IFS/3000 Compiled Character Cell file
1114	RASTR	Graphics Image in RASTR Format
1130	OPTLF	OPT/3000 Log file
1131	TEPES	TEPE/3000 Script file
1132	TEPEL	TEPE/3000 Log file
1133	SAMPL	APS/3000 Log file
1139	MPEDL	MPEDCP/DRP Log file
1140	TSR	HPToolset Root file
1141	TSD	HPToolset Data file
1145	DRAW	Drawing file for HPDRAW
1146	FIG	Figure file for HPDRAW
1156	DSTOR	Store File for RAPID/3000 utility DICTDBU
1157	TCODE	Code File for Transact/3000 Compiler
1158	RCODE	Code File for Report/3000 Compiler
1159	ICODE	Code File for Inform/3000 Compiler
1166	MDIST	HPMAIL Distribution list
1167	MTEXT	HPMAIL Text
1192	NCONF	Network Configuration file
1193	NTRAC	Network Trace file
1194	NLOG	Network Log file

HPMAIL Customer Training Goes ON-LINE!

by Marguerite Hebert/Information Networks Division

For the first time HP offers customer training that is completely on-line. "Learning HPMAIL" is an interactive training course that teaches how to use HP's electronic mail system. Any user with a terminal will be able to access the training course once it is installed.

"Learning HPMAIL" consists of a series of modules. This format provides both choice and flexibility. The Introductory module makes users fully functional on selected product basics. Additional modules cover specific HPMAIL functions, such as creating distribution lists, filing, and sending messages containing information from other HP3000 text-processing subsystems.

Interactive customer training for HPMAIL was produced using HP's own Interactive Training Facility (ITF). This facility was developed by instructional software specialists at Information Networks Division (IND). ITF includes an authoring language that allows HP course developers to write computer-based training courses for various products.

The interactive nature of "Learning HPMAIL" makes it an ideal learning tool. Users learn functions by doing exercises and completing activities that simulate using the product. The same menus, prompts, and messages that appear in HPMAIL are duplicated in the interactive training course. Users get a lot of practice and immediate feedback.

The price for the standard product (36570TA) is $700. The "right-to-copy without sublicense" product (36570TM) is $490. The standard tape format for this course is 1600 bpi magnetic tape, but it can be ordered on 7908/11/12 compatible cartridge tape by including Option 022 with either product number.

Contact your local Hewlett-Packard Sales Office for more information.

IFS/3000: Merging Text and Graphics

by Ellen Brigham, Information Networks Division

IFS/3000 has been enhanced to support graphics output both interactively and programmatically. A new program, the LPS Interpreter, allows a user to access the laser printer intrinsics interactively. Eight new intrinsics have been added to allow programmatic control of graphics output. Together, all these features comprise the HP 2680 Graphics Package (HP 36583A).

Changes have also been made to the standard character sets supplied with IFS/3000:

- The HELV character set has been updated to include a corrected 8 point and a new 10 point font.
- The new HELVBOLD character set contains 8, 14 and 24 point fonts.
- A 12 point math character set has been added.
- There are now Roman extension character sets for ELITE, LTITAL, PICA, ROM, and SCRIPT.
- ISO substitution character sets for Pica and Elite in French, German, Norwegian/Danish, Spanish, UK, and Swedish/Finnish, and environments which utilize the character sets, have been added.

Minimum Hardware Configuration

- HP 2680A Laser Printer.
- Installation Kit, option 300, 301, 302, 340, or 364.
- Graphics firmware (option 060) for new HP 2680As, or HP 26086A upgrade kit for installed HP 2680As.
- One megabyte of printer memory (option 020) for new HP 2680As, or HP 26085A memory for installed HP 2680As.

Minimum HP Graphics Software

o HP Printer Support Package (PSP/3000)

o At least one of the following:

  HP 3000 Business Graphics Package
  HPEASYCHART
  HPDRAW
  DSG/3000

Clarifying MPE Date Codes

by Brian B. Egan, Computer Systems Division

The MPE Documentation Group has had several requests for a clearer definition of the date codes for various versions of MPE. Beginning with this issue of the Communicator, all date codes will be expressed in a consistent way in all operating system documentation. The operating system appearing on the latest MIT will be referred to throughout as the "C/D.01.00 release of MPE".

Previously, the version of the operating system was expressed in one of three ways: the HP Code Name, the Date Code, and the *v.uu.ff*. We will be standardizing on the *v.uu.ff* in all future publications. It will appear on the Printing History page of all MPE manuals so that you may determine whether the manuals apply to the version of software you are currently using.

The following table cross-references the previous notations:

MIT Code Name	Date Code	*v.uu.ff*
ATHENA		B.01.01
BRUNO	2028	B.01.02
CHEETAH	2101	C.00.01
C-Delta	2134	C.00.03
D	2131	C.00.02
D-Delta	2146	C.00.04
CUB	2202	C.00.06
IMPACT	2206	C.00.08
CIPER	2226	C.00.20 (All except Series 64) D.00.20 (Series 64)
Q-MIT	2244	C.01.00 (All except Series 64) D.01.00 (Series 64)

HP 3000 Series II and Two-bay Series III Discontinuance

by Vance Ikezoye, Computer Systems Division

In today's dynamic environment, products and technologies change very quickly. Many of the components used in our older systems are no longer available to us. This, coupled with our commitment to offer you new products with increased performance at lower cost, has resulted in our decision to discontinue the Series II and the two-bay Series III. These systems, and the add-on and upgrade products uniquely associated with them, will be discontinued on February 1, 1983. Although these products will not be available for purchase after that date, full HP service and support will continue for five years, until February 1, 1988. For further information, contact your HP Sales Representative.

Identification of Files in System Account

compiled by Tom Peterson, Information Networks Division

ACCOUNT= SYS GROUP= CHARSETS

FILENAME	GROUP	ACCT	PRODUCT# VUF	DESCRIPTION
BAR39W14	CHARSETS	SYS	HP36580A.01.00	IFS/3000
BAR39W19	CHARSETS	SYS	HP36580A.01.00	IFS/3000
BAR39W25	CHARSETS	SYS	HP36580A.01.00	IFS/3000
BIGONES	CHARSETS	SYS	HP36580A.01.00	IFS/3000
COUR	CHARSETS	SYS	HP36580A.01.00	IFS/3000
COUR12	CHARSETS	SYS	HP36580A.01.00	IFS/3000
COURITAL	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITE	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEF	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEG	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEN	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITES	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEU	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEW	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ELITEX	CHARSETS	SYS	HP36580A.01.00	IFS/3000
HELV	CHARSETS	SYS	HP36580A.01.00	IFS/3000
HELVBOLD	CHARSETS	SYS	HP36580A.01.00	IFS/3000
HELVITAL	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LINEPR60	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LINEPR61	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LINEPR66	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LRGELITE	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LTITAL	CHARSETS	SYS	HP36580A.01.00	IFS/3000
LTITALX	CHARSETS	SYS	HP36580A.01.00	IFS/3000
MATH	CHARSETS	SYS	HP36580A.01.00	IFS/3000
OCRA	CHARSETS	SYS	HP36580A.01.00	IFS/3000
OCRB	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICA	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAF	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAG	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICALINE	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAN	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAS	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAU	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAW	CHARSETS	SYS	HP36580A.01.00	IFS/3000
PICAX	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ROM	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ROMBOLD	CHARSETS	SYS	HP36580A.01.00	IFS/3000
ROMITAL	CHARSETS	SYS	HP36580A.01.00	IFS/3000

```
ROMX             CHARSETS SYS          HP36580A.01.00             IFS/3000
SCRIPT           CHARSETS SYS          HP36580A.01.00             IFS/3000
SCRIPTX          CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCA20           CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCA25           CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCA35           CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCE20           CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCE25           CHARSETS SYS          HP36580A.01.00             IFS/3000
UPCE35           CHARSETS SYS          HP36580A.01.00             IFS/3000


ACCOUNT=  SYS         GROUP=  DOCUMENT

FILENAME    GROUP   ACCT     PRODUCT#   VUF  DESCRIPTION

MPEMIT2  DOCUMENT SYS                        MPE
NOONLST2 DOCUMENT SYS                        MPE


ACCOUNT=  SYS         GROUP=  ENV2680A

FILENAME         GROUP    ACCT         PRODUCT#    VUF            DESCRIPTION

BORDERS          ENV2680A SYS          HP36580A.01.00             IFS/3000
DOC              ENV2680A SYS          HP36580A.01.00             IFS/3000
DOC1X            ENV2680A SYS          HP36580A.01.00             IFS/3000
DOC2             ENV2680A SYS          HP36580A.01.00             IFS/3000
DOC2X            ENV2680A SYS          HP36580A.01.00             IFS/3000
DOC4             ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE            ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1F          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1G          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1N          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1S          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1U          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1W          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE1X          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2           ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2F          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2G          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2N          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2S          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2U          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2W          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE2X          ENV2680A SYS          HP36580A.01.00             IFS/3000
ELITE4           ENV2680A SYS          HP36580A.01.00             IFS/3000
GRAFLAND         ENV2680A SYS          HP36580A.01.00             IFS/3000
LAND             ENV2680A SYS          HP36580A.01.00             IFS/3000
LAND2            ENV2680A SYS          HP36580A.01.00             IFS/3000
LAND4            ENV2680A SYS          HP36580A.01.00             IFS/3000
LP               ENV2680A SYS          HP36580A.01.00             IFS/3000
LP2              ENV2680A SYS          HP36580A.01.00             IFS/3000
```

```
LP4                ENV2680A SYS         HP36580A.01.00            IFS/3000
LP60               ENV2680A SYS         HP36580A.01.00            IFS/3000
LP602              ENV2680A SYS         HP36580A.01.00            IFS/3000
LP604              ENV2680A SYS         HP36580A.01.00            IFS/3000
LTITAL1X           ENV2680A SYS         HP36580A.01.00            IFS/3000
LTITAL2X           ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA               ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1F             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1G             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1N             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1S             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1U             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1W             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA1X             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2              ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2F             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2G             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2N             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2S             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2U             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2W             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA2X             ENV2680A SYS         HP36580A.01.00            IFS/3000
PICA4              ENV2680A SYS         HP36580A.01.00            IFS/3000
PORT               ENV2680A SYS         HP36580A.01.00            IFS/3000
PORT2              ENV2680A SYS         HP36580A.01.00            IFS/3000
PORT4              ENV2680A SYS         HP36580A.01.00            IFS/3000
SCRIPT1X           ENV2680A SYS         HP36580A.01.00            IFS/3000


ACCOUNT=  SYS          GROUP=  FIGURE

FILENAME           GROUP    ACCT        PRODUCT#    VUF           DESCRIPTION

ARROWS             FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
DEVICES            FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
FLOWCHRT           FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
LETTERS            FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
LOGOS              FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
MAPS               FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
OFFICE             FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
PLANTS             FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
POLYGONS           FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
SHAPES             FIGURE   SYS         HP32108A.00.01            HPDRAW/3000
VEHICLES           FIGURE   SYS         HP32108A.00.01            HPDRAW/3000


ACCOUNT=  SYS          GROUP=  PUB

FILENAME         GROUP ACCT        PRODUCT#    VUF        DESCRIPTION

APL              PUB      SYS        HP32105A.01.07           APL/3000
ASOCTABL         PUB      SYS                                 MPE
```

BASIC	PUB	SYS	HP32101B.00.18	BASIC/3000
BASICOMP	PUB	SYS	HP32103B.00.18	BASICOMP/3000
BUILDINT	PUB	SYS		MPE
C00C180A	PUB	SYS	HP32180	SAMPLER
C01C108A	PUB	SYS	HP32108A.00.01	HPDRAW/3000
C01C250A	PUB	SYS	HP32250A.01.01	DSG/3000
C02C108A	PUB	SYS	HP32108A.00.01	HPDRAW/3000
C03C108A	PUB	SYS	HP32250A.01.01	DSG/3000
CATALOG	PUB	SYS		MPE
CATIMF	PUB	SYS	HP32229A.01.04	IMF/3000
CICAT	PUB	SYS		MPE
COBCAT	PUB	SYS	HP32233A.00.09	COBOLII/3000
COBCNV	PUB	SYS	HP32233A.00.09	COBOLII/3000
COBEDIT	PUB	SYS	HP32233A.00.09	COBOLII/3000
COBOL	PUB	SYS	HP32213C.02.11	COBOL
			HP32233A.00.09	COBOLII
COBOLII	PUB	SYS	HP32233A.00.09	COBOLII/3000
COMMAND	PUB	SYS		MPE
CONFDATA	PUB	SYS		MPE
CONVERT	PUB	SYS	HP32209B.03.03	VPLUS/3000
CSDBSC0	PUB	SYS	HP30130E.02.00	RJE
			HP32190A.04.04	DS DWNLD
CSDBSC1	PUB	SYS	HP30130E.02.00	RJE
			HP32190A.04.04	DS DWNLD
CSDBSC2	PUB	SYS	HP30130E.02.00	RJE
			HP32190A.04.04	DS DWLNLD
CSDBSCX0	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDBSCX1	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDBSCX2	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDIMF0	PUB	SYS	HP32229A.01.04	IMF/3000 INP DOWNLOAD
CSDIMFS0	PUB	SYS	HP32229A.01.04	IMF/3000 INP DOWNLOAD
CSDLAPB0	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDLAPB1	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDLAPB2	PUB	SYS	HP32190A.04.04	DS/3000 INP DOWNLOAD
CSDMRJE0	PUB	SYS	HP32192A.02.00	MRJE/3000 INP DOWNLOAD
CSDMRJE1	PUB	SYS	HP32192A.02.00	MRJE/3000 INP DOWNLOAD
CSDMRJE2	PUB	SYS	HP32192A.02.00	MRJE/3000 INP DOWNLOAD
CSDMTS0	PUB	SYS	HP32193A.04.32	MTS/3000 INP DOWNLOAD
CSDUMP	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
CSHBSC0	PUB	SYS	HP30131A.05.07	CS/3000 HSI DRIVER
CSLIST	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
CSSBSC0	PUB	SYS	HP30131A.05.07	CS/3000 SSLC DRIVER
CSSBSC1	PUB	SYS	HP32193A.04.32	MTS/3000
CSSMRJE0	PUB	SYS	HP32192A.02.00	MRJE/3000
DBDRIVER	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBDUMP	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBLOAD	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBRECOV	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBRESTOR	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBSCHEMA	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBSTORE	PUB	SYS	HP32215B.03.06	IMAGE/3000
DBUNLOAD	PUB	SYS	HP32215B.03.06	IMAGE/3000

DBUTIL	PUB	SYS	HP32215B.03.06	IMAGE/3000
DEVREC	PUB	SYS		MPE
DICTDBA	PUB	SYS	HP32244A.00.03	DICT/3000
DICTDBC	PUB	SYS	HP32244A.00.03	DICT/3000
DICTDBD	PUB	SYS	HP32244A.00.03	DICT/3000
DICTDBL	PUB	SYS	HP32244A.00.03	DICT/3000
DICTDBM	PUB	SYS	HP32244A.00.03	DICT/3000
DICTDBU	PUB	SYS	HP32244A.00.03	DICT/3000
DICTINIT	PUB	SYS	HP32244A.00.03	DICT/3000
DICTSTOR	PUB	SYS	HP32244A.00.03	DICT/3000
DISCCOPY	PUB	SYS	HP32199	DISCCOPY
DISKED2	PUB	SYS		MPE
DPAN4	PUB	SYS		MPE
DS2026	PUB	SYS	HP32190A.04.04	DS/3000
DS2026CN	PUB	SYS	HP32190A.04.04	DS/3000
DSCOPY	PUB	SYS	HP32190A.04.04	DS/3000
DSDUMP	PUB	SYS	HP32190A.04.04	DS/3000
DSM	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
DSMON	PUB	SYS	HP32190A.04.04	DS/3000
DSMONX	PUB	SYS	HP32190A.04.04	DS/3000
DSMTESTS	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
DSMTEXTS	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
DSTEST	PUB	SYS	HP32190A.04.04	DS/3000
DUMPARCH	PUB	SYS		MPE
DUMPJOB4	PUB	SYS		MPE
DUMPMINI	PUB	SYS		MPE
EDITOR	PUB	SYS	HP32201A.07.13	EDIT/3000
ENTRY	PUB	SYS	HP32209B.03.03	VPLUS/3000
EZCHART	PUB	SYS	HP32109A.00.00	HPEASYCHART/3000
F02F108A	PUB	SYS	HP32108A.00.01	HPDRAW/3000
F02F109A	PUB	SYS	HP32109A.00.00	HPEASYCHART/3000
F02F250A	PUB	SYS	HP32250A.01.01	DSG/3000
FCOPY	PUB	SYS	HP32212A.03.17	FCOPY/3000
FIGMAKER	PUB	SYS	HP32250A.01.01	DSG/3000
FORMAINT	PUB	SYS	HP32206A.01.10	DEL
FORMGEN	PUB	SYS		MPE
FORMSPEC	PUB	SYS	HP32209B.03.03	VPLUS/3000
FORTRAN	PUB	SYS	HP32102B.01.09	FORTRAN/3000
FREE2	PUB	SYS		MPE
FRMSPCFF	PUB	SYS	HP32209B.03.03	VPLUS/3000
GRAPH	PUB	SYS	HP32250A.01.01	DSG/3000
GUPDATE	PUB	SYS	HP32250A.01.01	DSG/3000
HIOFLOP0	PUB	SYS		MPE
HIOLPRT0	PUB	SYS		MPE
HIOLPRT1	PUB	SYS		MPE
HIOLPRT2	PUB	SYS		MPE
HIOMDSC1	PUB	SYS		MPE
HIOMDSC2	PUB	SYS		MPE
HIOPPRT0	PUB	SYS		MPE
HIOTAPE0	PUB	SYS		MPE
HIOTAPE1	PUB	SYS		MPE
HIOTERM0	PUB	SYS		MPE

HIOTERM1	PUB	SYS		MPE
HP7260A	PUB	SYS		MPE
HPDRAW	PUB	SYS	HP32108A.00.01	HPDRAW/3000
HPSLATE	PUB	SYS	HP36576A.01.01	HPSLATE
HPSLATEH	PUB	SYS	HP36576A.01.01	HPSLATE
HPSLATEP	PUB	SYS	HP36576A.01.01	HPSLATE
IDSCHAR	PUB	SYS	HP36581A.00.01	IDS/3000
IDSFORM	PUB	SYS	HP36581A.00.01	IDS/3000
IFS2680	PUB	SYS	HP36580A.01.00	IFS/3000
IMAGECAT	PUB	SYS	HP32215B.03.06	IMAGE/3000
INFCNV	PUB	SYS	HP32246A.00.03	INFORM/3000
INFORM	PUB	SYS	HP32246A.00.03	INFORM/3000
ININ	PUB	SYS		MPE
INITIAL	PUB	SYS		MPE
INPDPAN	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
IOCDPN0	PUB	SYS		MPE
IOCDRD0	PUB	SYS		MPE
IODS0	PUB	SYS	HP32190A.04.04	DS/3000
IODSTRM0	PUB	SYS	HP32190A.04.04	DS/3000
IODSTRMX	PUB	SYS	HP32190A.04.04	DS/3000
IODSX	PUB	SYS	HP32190A.04.04	DS/3000
IOFDISC0	PUB	SYS		MPE
IOINP0	PUB	SYS	HP30131A.05.07	CS/3000
IOLPRT0	PUB	SYS		MPE
IOM3270	PUB	SYS	HP32229A.01.04	IMF/3000
IOMCONS0	PUB	SYS		MPE
IOMDISC0	PUB	SYS		MPE
IOMDISC1	PUB	SYS		MPE
IOMPNLP0	PUB	SYS	HP32192A.02.00	MRJE/3000
IOMPS0	PUB	SYS	MP32193A.04.32	MTS/3000
IOMPTRM0	PUB	SYS	HP32193A.04.32	MTS/3000
IOMRDR0	PUB	SYS	HP32192A.02.00	MRJE/3000
IOMRJE0	PUB	SYS	HP32192A.02.00	MRJE/3000
IOMRJE1	PUB	SYS	HP32192A.02.00	MRJE/3000
IOPAD0	PUB	SYS	HP32190A.04.04	DS/3000
IOPLOT0	PUB	SYS		MPE
IOPRPN0	PUB	SYS		MPE
IOPTPN0	PUB	SYS		MPE
IOPTRD0	PUB	SYS		MPE
IOREM0	PUB	SYS		MPE
IOTAPE0	PUB	SYS		MPE
IOTERM0	PUB	SYS		MPE
IPCLEAN	PUB	SYS	HP32244A.00.03	DICT/3000
IPDIC	PUB	SYS	HP32244A.00.03	DICT/3000
KSAMUTIL	PUB	SYS	HP32208	KSAM
LINETEST	PUB	SYS		MPE
LISTDIR2	PUB	SYS		MPE
LISTEQ2	PUB	SYS		MPE
LISTLOG2	PUB	SYS		MPE
LOAD	PUB	SYS		MPE
LOADLIST	PUB	SYS		MPE
LOADMAP	PUB	SYS		MPE

LOG	PUB	SYS		MPE
LPS	PUB	SYS	HP36580A.01.00	IFS/3000
LPSCAT	PUB	SYS	HP36580A.01.00	IFS/3000
MAKECAT	PUB	SYS		MPE
MEMLOG	PUB	SYS		MPE
MEMLOGAN	PUB	SYS		MPE
MEMLOGP	PUB	SYS		MPE
MEMTIMER	PUB	SYS		MPE
MERGE	PUB	SYS	HP32214C.03.00	SORT/MERGE
MPCONFIG	PUB	SYS	HP32193A.04.32	MTS/3000
MPECHECK	PUB	SYS		MPE
MPMON	PUB	SYS	HP32193A.04.32	MTS/3000
MPTEST	PUB	SYS	HP32193A.04.32	MTS/3000
MRJE	PUB	SYS	HP32192A.02.00	MRJE/3000
MRJECAT	PUB	SYS	HP32192A.02.00	MRJE/3000
MRJELOGR	PUB	SYS	HP32192A.02.00	MRJE/3000
MRJEMON	PUB	SYS	HP32192A.02.00	MRJE/3000
MRJEOUT	PUB	SYS	HP32192A.02.00	MRJE/3000
NETCAT	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
NETCONF	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
NFTCAT	PUB	SYS	HP32190A.04.04	DS/3000
OPT	PUB	SYS	HP32238	OPT
P01P180A	PUB	SYS	HP32180	SAMPLER
P02P180A	PUB	SYS	HP32180	SAMPLER
P03P180A	PUB	SYS	HP32180	SAMPLER
PART2680	PUB	SYS	HP36580A.01.00	IFS/3000
PASCAL	PUB	SYS	HP32106A.00.05	PASCAL/3000
PASCAT	PUB	SYS	HP32106A.00.05	PASCAL/3000
PASLXREF	PUB	SYS	HP32106A.00.05	PASCAL/3000
PATCH	PUB	SYS		MPE
PFAIL	PUB	SYS		MPE
PMSGCAT	PUB	SYS	HP36580A.01.00 HP36581A.00.01	IFS IDS
PROGEN	PUB	SYS		MPE
PVINIT	PUB	SYS		MPE
PVPROC	PUB	SYS		MPE
QSMSGCAT	PUB	SYS	HP32216B.00.03	QUERY/3000
QUERY	PUB	SYS	HP32216B.00.03	QUERY/3000
RAPID	PUB	SYS	HP32249A.00.03	RAPID/3000
RAPIDCAT	PUB	SYS	HP32244A.00.03	RAPID/3000
RAPIDUDC	PUB	SYS	HP32249A.00.03	RAPID/3000
RECOVER2	PUB	SYS		MPE
REFORMAT	PUB	SYS	HP32209B.03.03	VPLUS/3000
REFSPEC	PUB	SYS	HP32209B.03.03	VPLUS/3000
REPCOMP	PUB	SYS	HP32245A.00.03	RAPID
REPORT	PUB	SYS	HP32245A.00.03	RAPID
RFMSPCFF	PUB	SYS	HP32209B.03.03	VPLUS/3000
RISE	PUB	SYS	HP32104A.05.06	RPG
RISEFORM	PUB	SYS	HP32104A.05.06	RPG
RISETOUR	PUB	SYS	HP32104A.05.06	RPG
RJE	PUB	SYS	HP30130E.01.00	RJE/3000
RPG	PUB	SYS	HP32104A.05.06	RPG

RPGMSGS	PUB	SYS	HP32104A.05.06	RPG
RTOKSAM	PUB	SYS	HP32208	KSAM
S20S109A	PUB	SYS	HP32109A.00.00	HPEASYCHART/3000
S21S109A	PUB	SYS	HP32109A.00.00	HPEASYCHART/3000
SADUTIL	PUB	SYS		MPE
SAMPLER	PUB	SYS	HP32180	SAMPLER
SEGDVR	PUB	SYS		MPE
SEGPROC	PUB	SYS		MPE
SEGSYM	PUB	SYS	HP32350A.00.01	HPTOOLSET
SIMCAL	PUB	SYS	HP32104A.05.06	RPG
SL	PUB	SYS		MPE
SLPATCH	PUB	SYS		MPE
SORT	PUB	SYS	HP32214C.03.00	SORT/MERGE
SPL	PUB	SYS	HP32100A.08.01	SPL/3000
SPLINTR	PUB	SYS		MPE
SPOOK	PUB	SYS		MPE
STORE	PUB	SYS		MPE
SUPACCT	PUB	SYS		MPE
SYSDUMP	PUB	SYS		MPE
TOOLSET	PUB	SYS	HP32350A.00.01	HPTOOLSET
TRACPROG	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
TRANCOMP	PUB	SYS	HP32247A.00.03	TRANSACT/3000
TRANSACT	PUB	SYS	HP32247A.00.03	TRANSACT/3000
TSETCAT	PUB	SYS	HP32350A.00.01	HPTOOLSET
TSETHELP	PUB	SYS	HP32350A.00.01	HPTOOLSET
TSETLABL	PUB	SYS	HP32350A.00.01	HPTOOLSET
TSETMEN1	PUB	SYS	HP32350A.00.01	HPTOOLSET
TSETMENU	PUB	SYS	HP32350A.00.01	HPTOOLSET
TSETUTIL	PUB	SYS	HP32350A.00.01	HPTOOLSET
TTSMON	PUB	SYS	HP32229A.01.04	IMF/3000
TTSSON	PUB	SYS	HP32229A.01.04	IMF/3000
TTSUSER	PUB	SYS	HP32229A.01.04	IMF/3000
UCOP	PUB	SYS		MPE
VECTRAST	PUB	SYS	HP36580A.01.00	IFS/3000
VERRMSGS	PUB	SYS	HP32209B.03.03	VPLUS/3000
VIDSCHAR	PUB	SYS	HP36581A.00.01	IDS/3000
VIDSFORM	PUB	SYS	HP36581A.00.01	IDS/3000
VIFS2680	PUB	SYS	HP36580A.01.00	IFS/3000
X21MSG	PUB	SYS	HP30131A.05.07	CS/3000 UTILITY
XA2100	PUB	SYS	HP32227	2100 CROSS LOADER
XL2100	PUB	SYS	HP32226	2100 CROSS LOADER
XRTCGEN	PUB	SYS	HP30301	XRTCGEN/3000
XSORT	PUB	SYS	HP32104A.05.06	RPG

ACCOUNT= SYS GROUP= VCHARSET

FILENAME	GROUP	ACCT	PRODUCT#	VUF	DESCRIPTION
GOTHIC	VCHARSET	SYS	HP32250A.01.01		DSG/3000
ROMAN	VCHARSET	SYS	HP32250A.01.01		DSG/3000
SCRIPT	VCHARSET	SYS	HP32250A.01.01		DSG/3000
STICK	VCHARSET	SYS	HP32250A.01.01		DSG/3000

Series II/III Software Update

```
              MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES II/III
********************************************************************************

                   CONTENTS OF INSTALLATION TAPE DATE CODE '2244'

  PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T..


********************************************************************************
 PRODUCT                    PRODUCT             LEVEL              DATE
 NAME                       NUMBER                                 CODE

*MPE                        32033C              01.00              2244
*SEGMENTER                  32050A              01.08              2244
 SPL                        32100A              08.01              2052
*BASIC                      32101B              00.18              2220
*FORTRAN                    32102B              01.09              2244
*BASIC COMPILER             32103B              00.18              2220
*RPG                        32104A              05.06              2244
 APL/3000                   32105A              01.07              2220
*PASCAL/3000                32106A              00.05              2244
*HPDRAW                     32108A              00.01              2244
*HPEASYCHART                32109A              00.00              2244
 HPWORD                     32120A              00.01              2220
 BUILDINT                   32150A              03.01              1623
*SAMPLER/3000               32180A              01.01              2244
*DS/3000                    32190A              04.04              2244
*MRJE                       32192A              02.00              2202
*MTS                        32193A              04.32              2244
*ATP                        32196D              01.00              2244
 DISCCOPY                   32199A              00.01              2052
*EDITOR                     32201A              07.13              2244
 SCIENTIFIC LIBRARY         32205B              00.05              2220
 DEL/3000                   32206A              01.10              2011
*KSAM/3000                  32208A              03.09              2244
*V/3000                     32209B              03.03              2244
*COMPILER LIBRARY           32211D              01.05              2244
*FCOPY                      32212A              03.17              2244
*COBOL                      32213C              02.11              2244
*SORT/MERGE                 32214C              03.00              2244
*IMAGE                      32215B              03.06              2244
*QUERY                      32216B              00.03              2244
 TRACE                      32222A              03.04              2222
*IMF                        32229A              01.04              2244
*DIAGNOSTICS                32231A              2236               2244
*COBOL LIBRARY              32232A              00.08              2244
*COBOLII                    32233A              00.09              2244
*OPT/3000                   32238A              00.10              2244
```

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*DATACAPTURE PROCEDURES	32243A	01.01	2244
*DICTIONARY/3000	32244A	00.03	2244
*RAPID/REPORT	32245A	00.03	2244
*RAPID/INFORM	32246A	00.03	2244
*PROG. PROD. PACK	32248A	00.03	2244
*RAPID/PROCESSOR	32249A	00.03	2244
*TRANSACT/3000	32247A	00.03	2244
*DSG/3000	32250A	01.01	2244
*DIAGNOSTICS	32340A	2241	2244
*HPTOOLSET	32350A	00.01	2244
*HPTOOLSET	32350A	00.01	2244
*HPSLATE	32576A	01.01	2244
*TDP/3000	36578A	03.00	2244
*IFS/3000	36580A	01.00	2244
IDS/3000	36581A	00.01	2131
*IFS/GRAPICS	36583A	01.00	2244
*PSP	36584A	01.00	2244
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701
RJE 2780/3780	30130E	01.00	2052
*CS	30131A	05.07	2244

MPE 32002C.01.00: MPEIV CHANGE HISTORY

MODULES MODIFIED C.01.00.XX

.00

- INITIAL
- SYSDUMP
- SEGPROC
- SEGDVR
- LOAD

- UCOP
- DEVREC
- PROGEN
- ININ
- LOG
- IOPTPNO
- IOTERMO
- HIOTERMO
- IOMDISC1
- PFAIL
- PVPROC
- VINIT
- HIOTAPEO
- HIOLPRTO
- HIOMDSC1
- HIOLPRT1
- HIOFLOPO
- MAKECAT
- FILEACC
- COMM'INT
- STORE/RESTORE
- ALLOCATE
- HARDRES
- ABORTDUMP
- MESSAGE
- PROCSEG
- NRIO
- PCREATE
- MORGUE
- BIPC
- IPC
- CHECKER
- UTILITY
- SEGUTIL
- LOADER1

.00(cont.)

- RINS
- DEBUG
- FIRMWARESIM
- SPOOLING
- SPOOLCOMS
- PVSYS
- UDC
- USER
- HELPUSER
- OPCOMMND
- SDISC
- MEASIO
- LOGSEGO
- LOGSEG1
- KERNELC
- KERNELD
- MISCSEGC
- MEASSEG
- FILEIO
- INCLPCB
- INCLMEAS
- INCLICS
- INCLHARD
- CATALOG
- CICAT

.01

- ININ
- HIOTERMO
- PFAIL
- HARDRES
- MORGUE
- SPOOLING
- OPCOMMND
- LOGSEGO
- LOGSEG1
- KERNELC
- KERNELD
- FILEIO
- INCLHARD
- CATALOG

.02

- INITIAL
- SYSDUMP
- UCOP
- DEVREC
- PROGEN
- ININ
- MEMLOGP
- IOPLOTO
- IOTAPEO
- HIOPPRTO
- SDFLOAD
- HIOTAPEO
- HIOMDSC1
- HIOFLOPO
- HIOTAPE1
- FILEACC
- COMM'INT
- STORE/RESTORE
- ALLOCATE
- HARDRES
- NRIO
- MORGUE
- JOBTABLE
- SPOOLING
- SPOOLCOMS
- OPCOMMND
- LABSEG
- KERNELC
- MEASSEG
- FILEIO
- INCLHARD
- CATALOG
- CICAT

MODULES MODIFIED FOR VERSION C.01.00.XX

.03

UCOP
DEVREC
PROGEN
ININ
PFAIL
FILEACC
ALLOCATE
HARDRES
ABORTDUMP
MESSAGE
NRIO
PCREATE
MORGUE
BIPC
LOADER1
DEBUG
NURSERY
USER
HELPUSER
OPCOMMND
LOGSEG0
LOGSEG1
KERNELC
FILEIO
CATALOG
CICAT

.04

INITIAL
SYSDUMP
SEGPROC
SEGDVR
DEVREC
PROGEN
ININ
IOFDISC0
IOTERM0
HIOTEM0
IOMDISC1
HIOTAPE1
COMM'INT
STORE/RESTORE
HARDRES
ABORTDUMP
MESSAGE
NRIO
UTILITY
SEGUTIL
LOADER1
NURSERY
KERNELC
KERNELD
MEASSEG
INCLPCB
INCLHARD
INCLMSG
CATALOG
CICAT

.05

SEGPROC
SEGDVR
HIOTEM0
HARDRES
NRIO
CATALOG
CICAT

.06

INITIAL
SYSDUMP
DEVREC
PROGEN
ININ
PFAIL
PVPROC
VINIT
SDFLOAD
HIOMDSC1
HIOCTAP0
HIOMDSC2
FILEACC
COMM'INT
STORE/RESTORE
ALLOCATE
DFS
HARDRES
MMDISKR
ATPINIT
NRIO
LPMON
TERMON
UTILITY
SPOOLCOMS
PVSYS
OPCOMMND
LABSEG
SDISC
PDMANAGR
LOGSEG0
LOGSEG1
KERNELD
ATPDRIVR
FILEIO
INCLHARD
INCLMSG
INCLFREE
INCLDFS1
INCLDFS2
INCLDFSC
INCDISC1
INCDISC2
SADTLRL
CATALOG
WCS

MODULES MODIFIED FOR VERSION C.01.00.XX

.07

GUS
SYSDUMP

.08

INITIAL
SYSDUMP
HIOCTAP0
ALLOCATE
HARDRES
CATALOG
INCLHARD
INCLFREE
PVPROC
VINIT
HIOTERM0
CI
HIOMDSC1
UDC
SDISC
LOGSEG0
FILEIO
HIOMDSC2
ATPINIT
LPMON
TERMMON
PDMANAGR
ATPDRIVR
HIOTERM1
HIOASLP0
SDFLOAD
RESTORE
DPAN4
FREE2 (NEW SOURCE ON MMT)
HIOTAPE1
KERNELC
WCS

.10

CATALOG
OPCOMMND
MORGUE
COMM'INT
SPOOLCOM
ALLOCATE
SPOOLING
PROGEN
ININ
NRIO
INCLHARD
FILEACC
SADTLRL
INCLVUF
LISTEQ2
SADUTIL
DPAN4
KERNELD
SPOOK
RECOVER2
PROCSEG
PVPROC
FILEIO
LOGSEG0
LOGSEG1
SEGPROC
KERNELC
RESTORE
SYSDUMP
VINIT
IPC
HIOLPRT2

.10(cont.)

SEGDVR
MEASSEG
HARDRES
IOMDISC1
MEASIO
INCLMIFT
INCLMEAS
ASOCTABL
DISKED2
FREE2
IOCDPN0
LISTDIR2
LISTLOG2
PATCH
MEMLOGAN
MEMTIMER
SLPATCH
SPOOK
GUS
WCS

MODULES MODIFIED FOR VERSION C.01.00.XX

.20

INCLMIFT
INCLVMC
LOADER
SEGDVR
FREE2
DISKED2
LISTDIR2
VINIT
KERNELC
KERNELD
INCLHARD
CATALOG
HIOTAPE1
RESTORE
INITIAL
FILEACC
HARDRES
MEASSEG
MEMTIMER
MAKECAT
MESSAGE
HIOLRPT2
HIOCIPRO
SPOOLING
FILEIO
SYSDUMP
RECOVER2
SPOOK
SEGPROC
HIOCIPRO
INCLDT
INCLDLDTX
INCLXDD

.20(cont.)

INCLGBL
INCLVDEV
NRIO
ALLOCATE
DISKEDT2
SADUTIL
LISTLOG2
LISTEQ2
LISTDIR2
GUS
OPCOMMND
DFS
SPOOLCOMS
SPOOLING
SOFTIO
HIOMDSC2
SDFGEN
SDFLOAD
DEBUG
MEASIO
CHECKER
FIRMWARE
SLPATCH

.01.00

INCLCIS
UCOP
PCREATE
MORGUE
UDC
OPCOMMAND
LOADER1
INCLFREE
KERNELD
FILEIO
LABSEG
HARDRESS(BOTH)
FREE2
FILEACC
CIFILES
CIMAIN
CIMGR
CICAT
LISTLOG2
VINIT
INITIAL
SYSDUMP
NURSERY
LOGSEG1
SEGPROC
SEGDVR
SEGUTIL
CHECKER
UTILITY
USER
ININ(BOTH)
MAKECAT
JOBTABLE
ABORTDUMP
ALLOCATE
LOG
SOFTIO
SLPATCH
DPAN4
PATCH
INCPCB
INCLSLL

.01.00 (cont.)

PVPROC
PROGEN
MISCSEGC
SPOOLING
MEASEG
INCLHARD
INCLPCB
CATALOG
DFS
RESTORE
STORE
DIRC
INITDIR
SPOOOLCOMS
HIOMDISC2
DISKED2
LOAD
RINS
LABSEG
HIOCDRO
DEVREC
ASOCTABL
HIOTERMO
HIOFLOPO
SDISC
SADUTIL
IOCDPNO
SADTLRL
HIOTAPEO
TERMRES
INCLRIT
PVSYS
MEMTIMER
MEMLOGAN
IOCDPNO
IPC
NRIO
MESSAGE
INCLVUF
INCLLOG
HIOCTAPEO
MEASIO

SYSTEM	LAST CHANGE NUMBER
B.00.00	0066
B.00.01	0134
B.00.02	0472
B.01.00	0789
B.01.01	1261
B.01.02	793-797,1283-1299,1400-1499
C.00.00	2056
C.00.01	2097
C.00.02	2730
C.00.03	2380
C.00.04	2873
C.00.05	4016
C.00.06	3689
C.00.08	3766
C.00.07	3727
C.00.10	4290
C.00.20	4393
C.01.00	5008

NOTE: Each change made to MPE is now identified by a unique change number in columns 64/72 (eg <<00120>>). This matrix provides the range of the change numbers used to build each version of MPE.

HP-IB Software Update

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES 44/64

****

CONTENTS OF INSTALLATION TAPE DATE CODE '2244'

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T..

****

PRODUCT	PRODUCT		DATE
NAME	NUMBER	LEVEL	CODE
*MPE	32033C	01.00	2244
*SEGMENTER	32050A	01.08	2244
SPL	32100A	08.01	2052
*BASIC	32101B	00.18	2220
*FORTRAN	32102B	01.09	2244
*BASIC COMPILER	32103B	00.18	2220
*RPG	32104A	05.06	2244
*PASCAL/3000	32106A	00.05	2244
*HPDRAW	32108A	00.01	2244
*HPEASYCHART	32109A	00.00	2244
HPWORD	32120A	00.01	2220
BUILDINT	32150A	03.01	1623
*SAMPLER/3000	32180A	01.01	2244
*DS/3000	32190A	04.04	2244
*MRJE	32192A	02.00	2244
*MTS	32193A	04.32	2244
*ATP	32196D	01.00	2244
DISCCOPY	32199A	00.01	2052
*EDITOR	32201A	07.13	2244
SCIENTIFIC LIBRARY	32205B	00.05	2220
DEL/3000	32206A	01.10	2011
*KSAM/3000	32208A	03.09	2244
*V/3000	32209B	03.03	2244
*COMPILER LIBRARY	32211D	01.05	2244
*FCOPY	32212A	03.17	2244
*COBOL	32213C	02.11	2244
*SORT/MERGE	32214C	03.00	2244
*IMAGE	32215B	03.06	2244
*QUERY	32216A	00.03	2244
TRACE	32222A	03.04	2220
*IMF	32229A	01.04	2244
*DIAGNOSTICS	32231A	2236	2244
*COBOL LIBRARY	32232A	00.08	2244
*COBOLII	32233A	00.09	2244
*OPT/3000	32238A	00.10	2244
*DATACAPTURE PROCEDURES	32243A	01.01	2244

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*DICTIONARY/3000	32244A	00.03	2244
*RAPID/REPORT	32245A	00.03	2244
*RAPID/INFORM	32246A	00.03	2244
*TRANSACT/3000	32247A	00.03	2244
*PROG. PROD. PACK	32248A	00.03	2244
*RAPID/PROCESSOR	32249A	00.03	2244
*DSG/3000	32250A	01.01	2244
*DIAGNOSTICS	32340A	2241	2244
*DIAGNOSTICS	32342A	0000	2244
*HPTOOLSET	32350A	00.01	2244
*HPSLATE	32576A	01.01	2244
*TDP/3000	36578A	03.00	2244
*IFS/3000	36580A	01.00	2244
IDS/3000	36581A	00.01	2131
*IFS-GRAPICS	36583A	01.00	2244
*PSP	36584A	01.00	2244
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701
RJE 2780/3780	30130E	01.00	2052
*CS	30131A	05.07	2244

MPE HP32033 C/D.01.00.XX

MPE 32033 C/D.01.00.XX MPEIV CHANGE HISTORY

MODULES MODIFIED FOR VERSION C/D.01.00.XX

.00

INITIAL
SYSDUMP
SEGPROC
SEGDVR
LOAD
UCOP
DEVREC
PROGEN
ININ
LOG
IOPTPN0
IOTERM0
HIOTERM0
IOMDISC1
PFAIL
PVPROC
VINIT
HIOTAPE0
HIOLPRT0
HIOMDSC1
HIOLPRT1
HIOFLOP0
MAKECAT
FILEACC
COMM'INT
STORE/RESTORE
ALLOCATE
HARDRES
ABORTDUMP
MESSAGE
PROCSEG
NRIO

.00(cont.)

PCREATE
MORGUE
BIPC
IPC
CHECKER
UTILITY
SEGUTIL
LOADER1
RINS
DEBUG
FIRMWARESIM
SPOOLING
SPOOLCOMS
PVSYS
UDC
USER
HELPUSER
OPCOMMND
SDISC
MEASIO
LOGSEG0
LOGSEG1
KERNELC
KERNELD
MISCSEGC
MEASSEG
FILEIO
INCLPCB
INCLMEAS
INCLICS
INCLHARD
CATALOG
CICAT

.01

ININ
HIOTERM0
PFAIL
HARDRES
MORGUE
SPOOLING
OPCOMMND
LOGSEG0
LOGSEG1
KERNELC
KERNELD
FILEIO
INCLHARD
CATALOG

.02

INITIAL
SYSDUMP
UCOP
DEVREC
PROGEN
ININ
MEMLOGP
IOPLOT0
IOTAPE0
HIOPPRT0
SDFLOAD
HIOTAPE0
HIOMDSC1
HIOFLOP0
HIOTAPE1
FILEACC
COMM'INT
STORE/RESTORE
ALLOCATE
HARDRES
NRIO
MORGUE
JOBTABLE
SPOOLING
SPOOLCOMS
OPCOMMND
LABSEG
KERNELC
MEASSEG
FILEIO
INCLHARD
CATALOG
CICAT

MODULES MODIFIED FOR VERSION C/D.01.00.XX

.03

UCOP
DEVREC
PROGEN
ININ
PFAIL
FILEACC
ALLOCATE
HARDRES
ABORTDUMP
MESSAGE
NRIO
PCREATE
MORGUE
BIPC
LOADER1
DEBUG
NURSERY
USER
HELPUSER
OPCOMMND
LOGSEG0
LOGSEG1
KERNELC
FILEIO
CATALOG
CICAT

.04

INITIAL
SYSDUMP
SEGPROC
SEGDVR
DEVREC
PROGEN
ININ
IOFDISC0
IOTERM0
HIOTEM0
IOMDISC1
HIOTAPE1
COMM'INT
STORE/RESTORE
HARDRES
ABORTDUMP
MESSAGE
NRIO
UTILITY
SEGUTIL
LOADER1
NURSERY
KERNELC
KERNELD
MEASSEG
INCLPCB
INCLHARD
INCLMSG
CATALOG
CICAT

.05

SEGPROC
SEGDVR
HIOTEM0
HARDRES
NRIO
CATALOG
CICAT

.06

INITIAL
SYSDUMP
DEVREC
PROGEN
ININ
PFAIL
PVPROC
VINIT
SDFLOAD
HIOMDSC1
HIOCTAP0
HIOMDSC2
FILEACC
COMM'INT
STORE/RESTORE
ALLOCATE
DFS
HARDRES
MMDISKR
ATPINIT
NRIO
LPMON
TERMON
UTILITY
SPOOLCOMS
PVSYS
OPCOMMND
LABSEG
SDISC
PDMANAGR
LOGSEG0
LOGSEG1
KERNELD
ATPDRIVR
FILEIO
INCLHARD
INCLMSG
INCLFREE
INCLDFS1
INCLDFS2
INCDISC1
INCDISC2
SADTLRL
CATALOG
WCS

MODULES MODIFIED FOR VERSION C/D.01.00.XX

.07

GUS
SYSDUMP

.08

INITIAL
SYSDUMP
FREE2(NEW SOURCE ON MMT)
HIOCTAPO
ALLOCATE
HARDRES
CATALOG
INCLHARD
D326
INCLFREE
PVPROC
VINIT
HIOTERMO
CI
HIOMDSC1
UDC
SDISC
LOGSEGO
FILEIO
HIOMDSC2
SDFLOAD
RESTORE
DPAN4
HIOTAPE1
KERNELC
WCS
ATPINIT
HIOTERM1
HIOASLPO
LPMON
TERMON
ATPDRIVR
PDMANAGR

.10

CATALOG
OPCOMMND
MORGUE
COMM'INT
SPOOLCOM
ALLOCATE
SPOOLING
PROGEN
ININ
NRIO
INCLHARD
FILEACC
SADTLRL
INCLVUF
LISTEQ2
SADUTIL
DPAN4
KERNELD
SPOOK
RECOVER2
PROCSEG
PVPROC
FILEIO
LOGSEGO
LOGSEG1
SEGPROC
KERNELC
RESTORE
SYSDUMP
VINIT
IPC
HIOLPRT2
SEGDVR

.10(cont.)

MEASSEG
HARDRES
IOMDISC1
MEASIO
INCLMIFT
ASOCTABL
INCLMEAS
DISKED2
FREE2
IOCDPNO
LISTDIR2
LISTLOG2
PATCH
MEMLOGAN
MEMTIMER
SLPATCH
SPOOK
GUS
WCS

MODULES MODIFIED FOR VERSION C/D.01.00.XX

.20

INCLMIFT
INCLVMC
LOADER
SEGDVR
FREE2
DISKED2
LISTDIR2
VINIT
KERNELC
KERNELD
INCLHARD
CATALOG
HIOTAPE1
RESTORE
INITIAL
FILEACC
HARDRES
MEASSEG
MEMTIMER
MAKECAT
MESSAGE
HIOLRPT2
HIOCIPRO
SPOOLING
FILEIO
SYSDUMP
RECOVER2
SPOOK
SEGPROC
HIOCIPRO
INCLDT
INCLDLDTX
INCLXDD

.20(cont.)

INCLGBL
INCLVDEV
NRIO
ALLOCATE
DISKEDT2
SADUTIL
LISTLOG2
LISTEQ2
LISTDIR2
GUS
OPCOMMND
DFS
SPOOLCOMS
SPOOLING
SOFTIO
HIOMDSC2
SDFGEN
SDFLOAD
DEBUG
MEASIO
CHECKER
FIRMWARE
SLPATCH

01.00

INCLCIS
UCOP
PCREATE
MORGUE
UDC
OPCOMMAND
LOADER1
INCLFREE
KERNELD
FILEIO
LABSEG
HARDRESS(BOTH)
FREE2
FILEACC
CIFILES
CIMAIN
CIMGR
CICAT
LISTLOG2
VINIT
INITIAL
SYSDUMP
NURSERY
LOGSEG1
SEGPROC
SEGDVR
SEGUTIL
CHECKER
UTILITY
USER
ININ(BOTH)
MAKECAT
JOBTABLE
ABORTDUMP
ALLOCATE
LOG
SOFTIO
SLPATCH
DPAN4
PATCH
INCPCB
INCLSLL

.01.00(cont.)

PVPROC
PROGEN
MISCSEGC
SPOOLING
MEASEG
INCLHARD
INCLPCB
CATALOG
DFS
RESTORE
DIRC
INITDIR
SPOOOLCOMS
HIOMDISC2
DISKED2
LOAD
RINS
LABSEG
HIOCDRO
DEVREC
ASOCTABL
HIOTERMO
HIOFLOPO
SDIDC
ASDUTIL
IOCDPNO
SADTLRL
HIOTAPEO
TERMRES
INCLRIT
PVSYS
MEMTIMER
MEMLOGAN
IOCDPNO
IPC
NRIO
MESSAGE
INCLVUF
INCLLOG
HIOCTAPEO
MEASIO

SYSTEM	LAST CHANGE NUMBER
B.00.00	0066
B.00.01	0134
B.00.02	0472
B.01.00	0789
B.01.01	1261
B.01.02	793-797,1283-1299,1400-1499
C.00.00	2056
C.00.01	2097
C.00.02	2730
C.00.03	2380
C.00.04	2873
C.00.05	4016
C.00.06	3689
C.00.08	3766
C.00.07	3727
C.00.10	4290
C.00.20	4393
C.01.00	5008

NOTE: Each change made to MPE is now identified by a unique change number in columns 64/72 (eg <<00120>>). This matrix provides the range of the change numbers used to build each version of MPE.

New Or Changed Documentation

by Documentation Groups, Computer Systems and Information Networks Divisions

MPE Commands Reference Manual (30000-90009)
Update 3, November 1982 (U1182)

This update reflects all 2244 release changes, and all corrections and enhancements resulting from Service Requests.

MPE Segmenter Reference Manual (30000-90011)
Third Edition, November 1982 (E1182)

For this new edition the MPE Segmenter Reference Manual has been rewritten to include more information, and reorganized to make the information easier to find and use. The manual now provides more background on what the Segmenter is and how it works, and discusses strategies for effective Segmenter use. An intrinsic and several previously undocumented, are included. The Error Messages Table has more detailed comments and suggests actions to aid in recovery.

DSN/RJE 2780/3780 Emulator Reference Manual (30000-90047)
Fourth Edition, February 1982 (E0282)

This new edition describes the features incorporated in 30130E.2.00; automatic dial-up connection using DSN/INP, access to X.21, and programmatic control. Also documented are the subsystem commands that have been added or modified for programmatic control mode. These are: RJCMDFILE, RJCOMMENT, RJCONTINUE, RJLINE, RJOUT, and RJCONTINUE.

MPE System Utilities Reference Manual (30000-90044)
Third Edition, November 1982 (E1182)

The Utilities Manual is greatly improved in completeness, accuracy, style, and appearance. It reflects software enhancements and fixes, and corrections to the manual itself in response to reader comments and Service Requests.

HPEASYCHART Reference Manual (32109-90001)
First Edition, August 1982 (E0882)

This manual documents a new software subsystem which aids the inexperienced user to produce charts. HPEASYCHART is ideal for one time only chart creation, but charts made by HPEASYCHART may be accessed by DSG/3000 for additional enhancements. Also, the creation of figure files allows access by other subsystems, such as HPDRAW, TDP/3000, and HP2680A (Laser Printing System). The manual includes a brief, self-paced course as an introduction to HPEASYCHART.

DSN/DS for HP 3000 to HP 3000 Reference Manual (32190-90001)
Second Edition, September 1982 (E0982)

This manual (previously titled "DS/3000 Reference Manual") incorporates all updates as well as some new material. The Network File Transfer (NFT) documentation has been reformatted as Section VI. Section VII, "DS Applications", has been added to document the advanced uses and networking possibilities available with the enhanced DSN/DS. Appendix H documents the X.25 Network Configurator and Appendix J is a commentary on using Public Data Networks (PDNs); both are new. Also, Appendix A, Configuration Dialogue, has been updated to include the new driver names required for the utilization of DS/X.25.

DSN/MRJE Multileaving Remote Job Entry Reference Manual (32192-90001)
Fifth Edition, March 1982 (E0382)

This edition describes the new features incorporated in 32192A.02.00: The DSN/MRJE monitor is now a user process. One benefit is to allow a special form of file equations. Solicited job output may be routed to unsolicited output devices. Infiles can be submitted in transparent mode. Automatic dial up connection can take place between an HP 3000 using a DSN/INP and a host. DSN/MRJE subsystem messages are now in their own catalog file. And finally, a HELP command and a performance message facility to measure host job data received have been added.

DSN/MTS Multipoint Terminal Software Reference Manual (32193-90002)
Fifth Edition, August 1982 (E0882)

A number of enhancements to the DSN/MTS product are described in this new edition. DSN/MTS now supports the HP 2608S as a local or remote-spooled printer. The program for building configuration files, MPCONFIG.PUB.SYS, has been modified to include the HP 2608S and the multipoint testing program, MPTEST.PUB.SYS, has new capabilities. In Section 2 of the manual, the discussion of the FCONTROL intrinsic has been rewritten. Appendix B originally contained modem strapping information, has been deleted, as it is available in the Communications Handbook (30000-90105).

Data Entry and Forms Management System (VPLUS/3000) Reference Manual
(32209-90001)
Fourth Edition, October 1982

The fourth edition of the VPLUS/3000 Reference Manual documents local form storage on the HP 2624B terminal. Appendix G has been rewritten to provide more usable information about the terminals VPLUS/3000 supports. Various technical and editorial corrections have also been incorporated. Section VII, which described the RPG interface to VPLUS/3000, has been removed and placed in the RPG/3000 reference manual.

Dictionary/3000 Reference Manual (32244-90001)
Second Edition, July 1982

The second edition of the Dictionary Reference Manual documents using the DICTDBM commands to define HP Inform/3000 groups. A new appendix provides additional information on how HP Inform/3000 links files to generate reports. In addition, a Glossary of Terms and a DICTDBM Quick Reference have been added. Minor corrections and additions have been made throughout the manual.

Report/3000 User's Guide (32245-90001)
May 1982

This new user's guide describes the use of HP Report/3000, a command driven, non-procedural report writer used with HP Dictionary/3000 or as a standalone product. The first half of the guide is tutorial. The remainder is designed to be used as a reference document. Appendixes include error messages with explanations, a glossary of terms, and a quick-reference guide.

Transact/3000 Reference Manual (32247-90001)
Second Edition, October 1982

The second edition of the Transact/3000 Reference Manual documents the new features of Version A.00.01 (the June 1982 release). In addition, the text has been expanded and improved to explain concepts such as "alias" items and the STATUS option. More Transact programs examples have been added. The format of the verb section has been changed to improve usability. Section VII, Transact Test Facility, now contains examples of the different test modes. The flow charts of data base and file operation verbs are more complete.

Decision Support Graphics/3000 (DSG/3000) Manual (32250-90001)
Second Edition, August 1982

The second edition of the Decision Support Graphics/3000 Manual documents several enhancements which add to the power and flexibility of the subsystem. Increased text control allows the user to specify fonts and text sizes. Figure files can be created which are accessible to other subsystems, such as TDP/3000, HPDRAW, and the HP 2680A (Laser Printing System). Seven languages may be specified for use in chart development which include special characters for foreign languages. New devices have been added to the list of those supported by DSG/3000.

HPToolset Reference Manual (32350-90001)
July, 1982

The HPToolset Reference Manual documents HPToolset, a set of integrated development tools for programmers. HPToolset Reference Manual assumes a knowledge of COBOL and the HP 3000. Each of the Toolset 'tools' is described. These include: Workspace Manager, the Toolset full screen Editor, the Interface Program Key to COBOL II, and the Symbolic Debug facility. Also included in the manual is a tutorial section that illustrates how Toolset improves programmer efficiency.

New Course:

Using Dictionary/3000 Course
Hewlett-Packard Self-Paced Learning Series
Product Number 22843A
December 1981

Using Dictionary/3000 is a self-paced introduction to the HP data dictionary facility, aimed at the non-technical user. As an interactive course, it leads the student through the steps of defining entries in the data and of creating and maintaining IMAGE/3000 data bases. This course also provides background information on IMAGE, MPE, KSAM, and VPLUS files, to give the user an overview of the types of files they will be documenting in the data dictionary.

```
    XXXX        XX    XXXXXXXX    XX      XX         XXXX        XXXX
  XX    XX     XXXX      XX      XXXX     XX       XX    XX    XX    XX
 XX           XX  XX     XX     XX  XX    XX      XX      XX  XX
 XX          XX    XX    XX    XX    XX   XX      XX      XX  XX
 XX          XXXXXXXX    XX    XXXXXXXX   XX      XX      XX  XX  XXXXX
  XX    XX  XX      XX   XX   XX      XX  XX       XX    XX    XX    XX
    XXXX    XX      XX   XX   XX      XX  XXXXXXX    XXXX        XXXX

OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOO    OOOOOOOO  OOOO        OOOO  OOOOOO  OOOOOOOOO    OOOOOOOO    OOO
OO  OOOO  OOOOO    OOOOOO  OOOOOO    OOOOO  OOOOOOO  OOOO  OOOO  OOOO  O
O  OOOOOOOOOOO  OO  OOOOO  OOOOO  OO  OOOO  OOOOOO  OOOOOO  OO  OOOOOOOO
O  OOOOOOOOOO  OOOO  OOOO  OOOO  OOOO  OOO  OOOOOO  OOOOOO  OO  OOOOOOOO
O  OOOOOOOOOO        OOOO  OOOO        OOO  OOOOOO  OOOOOO  OO  OO     O
OO  OOOO  OO  OOOOOO  OOO  OOO  OOOOOO  OO  OOOOOOO  OOOO  OOOO  OOOO  O
OOOO    OOOO  OOOOOO  OOO  OOO  OOOOOO  OO       OOOO    OOOOOOOO    OOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

    XXXX        XX    XXXXXXXX    XX      XX         XXXX        XXXX
  XX    XX     XXXX      XX      XXXX     XX       XX    XX    XX    XX
 XX           XX  XX     XX     XX  XX    XX      XX      XX  XX
 XX          XX    XX    XX    XX    XX   XX      XX      XX  XX
 XX          XXXXXXXX    XX    XXXXXXXX   XX      XX      XX  XX  XXXXX
  XX    XX  XX      XX   XX   XX      XX  XX       XX    XX    XX    XX
    XXXX    XX      XX   XX   XX      XX  XXXXXXX    XXXX        XXXX

OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOO    OOOOOOOO  OOOOO      OOOOO  OOOOOO  OOOOOOOOO    OOOOOOOO    OOO
OO  OOOO  OOOOO    OOOOOO  OOOOOO    OOOOO  OOOOOOO  OOOO  OOOO  OOOO  O
O  OOOOOOOOOOO  OO  OOOOO  OOOOO  OO  OOOO  OOOOOO  OOOOOO  OO  OOOOOOOO
O  OOOOOOOOOO  OOOO  OOOO  OOOO  OOOO  OOO  OOOOOO  OOOOOO  OO  OOOOOOOO
O  OOOOOOOOOO        OOOO  OOOO        OOO  OOOOOO  OOOOOO  OO  OO     O
OO  OOOO  OO  OOOOOO  OOO  OOO  OOOOOO  OO  OOOOOOO  OOOO  OOOO  OOOO  O
OOOO    OOOO  OOOOOO  OOO  OOO  OOOOOO  OO       OOOO    OOOOOOOO    OOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

    XXXX        XX     XXXXXX     XX      XX         XXXX        XXXX
  XX    XX     XXXX      XX      XXXX     XX       XX    XX    XX    XX
 XX           XX  XX     XX     XX  XX    XX      XX      XX  XX
 XX          XX    XX    XX    XX    XX   XX      XX      XX  XX
 XX          XXXXXXXX    XX    XXXXXXXX   XX      XX      XX  XX  XXXXX
  XX    XX  XX      XX   XX   XX      XX  XX       XX    XX    XX    XX
    XXXX    XX      XX   XX   XX      XX  XXXXXXX    XXXX        XXXX
```

KEY

Manuals that are new or have changed since the last edition of this catalog are noted by an asterisk (*) in the Edition Date column. The Subscription Service column indicates the external subscription service on which the manual may be obtained. FOS indicates that the manual is available on all of the Fundamental Operating Software Subscription Services. A number in the Subscription Service column indicates the specific Subsystem Subscription Service on which the manual is listed.

DATA COMMUNICATIONS MANUALS

Manual Title	Part Number	Edition Date	Up-dated	Subscription Services
DSN/RJE 2780/3780 Emulator Ref. Man.	30000-90047	2/82*		30130Q,S,T
Communications Handbook	30000-90105	4/81		30130Q,S,T 32190Q,S,T
DSN/DS HP3000 to HP3000 Reference Manual	32190-90001	9/82*		32190Q,S,T
DSN/DS HP3000 to HP1000 Reference Manual for HP 3000 Users	32190-90005	1/82		32190Q,S,T
DSN/MRJE Multileaving Remote Job Entry Reference Manual	32192-90001	3/82*		32192Q,S,T
DSN/MTS Multipoint Terminal Software Reference Manual	32193-90002	8/81*		32193Q,S,T
DSN/IMF Interactive Mainframe Facility Reference Manual	32229-90001	9/81		32229Q,S,T
DSN/ATP On-Line Diagnostic Manual	30144-90003	2/82		

SYSTEM MANUALS

Manual Title	Part Number	Edition Date	Up-dated	Subscription Service
Using the HP 3000: An Introduction to Interactive Programming	03000-90121	4/79		
MPE Commands Reference Manual	30000-90009	1/81*	12/82	FOS, 30093Q
MPE Intrinsics Reference Manual	30000-90010	1/81	12/81	FOS, 30093Q
MPE Segmenter Reference Manual	30000-90011	11/82*		FOS, 30093Q
MPE Debug/Stack Dump Reference Manual	30000-90012	9/76	10/80	FOS, 30093Q
System Manager/System Supervisor Ref. Manual	30000-90014	12/81*	12/82	FOS
Error Messages and Recovery Manual	30000-90015	2/79		FOS, 300930
MPE System Utilities Reference Manual	30000-90044	11/82*		FOS, 300930
Index to MPE Reference Documents	30000-90045	5/81		FOS
Software Pocket Guide	30000-90049	1/81	4/81	FOS,30093Q
Using Files	30000-90102	4/78		FOS,30093Q
MPE File System Reference Manual	30000-90236	2/82		FOS
Console Operator's Guide	32002-90004	1/82*	12/82	FOS
Flexible Disccopy/3000	32199-90001	8/80		

SUBSYSTEM MANUALS

Manual Title	Part Number	Edition Date	Up-dated	Subscription Service
EDIT Reference Manual	03000-90012	8/80		FOS,30093Q
Trace Reference Manual	03000-90015	6/76		
FCOPY Reference Manual	03000-90064	7/80		FOS,30093Q
Scientific Library Reference Manual	30000-90027	6/76	2/77	32205Q,S,T
Compiler Library Reference Manual	30000-90028	11/76		30093Q,S,T
Flexible Discopy/3000	32199-90001	8/80		FOS,32199Q
APS/3000 Reference Manual	32180-90001	6/82		32180Q,S,T
APS/3000 Pocket Guide	32180-90003	6/82		32180Q,S,T
SORT Reference Manual	32214-90002	9/81		FOS,30093Q
OPT/3000 Reference Manual	32238-90001	8/81		32380Q,S,T

EDUCATIONAL APPLICATION MANUALS

Manual Title	Part Number	Edition Date	Up-dated	Subscription Service
Student Information System Reference Man.	32900-90001	9/74	8/76	32900Q,S,T
Student Information System Technical Man.	32900-90005	3/75		32900Q,S,T
Student Assignment System Reference Manual	32901-90001	8/78		32900Q,S,T 32901Q,S,T
Student Assignment System Technical Manual	32901-90005	8/78		32900Q,S,T 32901Q,S,T
College Information System Reference Man.	32902-90003	1/78		32902Q,S,T
College Information System Technical Man.	32902-90005	2/78		32902Q,S,T

LANGUAGE MANUALS

		Edition	Up-	Subscription
Manual Title	Part Number	Date	dated	Service
BASIC for Beginners	03000-90025	11/72		32111Q,S,T
BASIC/3000 Pocket Guide	03000-90050	9/74		32111Q,S,T
SPL Reference Man.	30000-90024	9/76	2/77	32100Q,S,T
SPL Language Textbook	30000-90025	6/76	1/77	32100Q,S,T
BASIC Interpreter Manual	30000-90026	6/76	8/78	30098Q,S,T 32111Q,S,T
FORTRAN Reference Manual	30000-90040	6/76	5/79	30099Q,S,T 32102Q,S,T
SPL Pocket Guide	32100-90001	11/76		32100Q,S,T
FORTRAN Pocket Guide	32102-90002	5/79		32102Q,S,T
BASIC Compiler Ref. Man.	32103-90001	11/74	6/76	32111Q,S,T
RPG/3000 Compiler Reference Manual	32104-90001	2/77	5/80	32104Q,S,T
RPG Listing Analyzer	32104-90003	2/77		32104Q,S,T
RPG Utilities Reference Manual	32104-90006	10/81		32104Q,S,T
APL Reference Manual	32105-90002	1/79		32105Q,S,T
APL Pocket Guide	32105-90003	11/76		32105Q,S,T
PASCAL Reference Man.	32106-90001	2/81		32106Q,S,T
COBOL Reference Manual	32213-90001	7/75	1/79	30096Q,S,T 32213Q,S,T
Using COBOL: A Guide for the COBOL Programmer	32213-90003	3/78		30096Q,S,T 32213Q,S,T
COBOL/II Reference Man.	32233-90001	12/79	7/80	32233Q,S,T
COBOL/3000 to COBOL II/3000 Conversion Guide	32233-90005	12/79		32233Q,S,T
HPToolset Ref. Man.	32350-90001	7/82*		32350Q,S,T

SOFTWARE PRODUCTS

Manual Title	Part Number	Edition Date	Up-dated	Subscription Service
DSG/3000 Manual	32250-90001	1/80*	8/82	32250Q,S,T
DSG/3000 Guide	32250-90002	1/80		32250Q,S,T
HPSLATE Reference Manual	36576-90001	8/81		36576Q,S,T
Using HPSLATE	36576-90002	8/81		32108Q,S,T
HPDRAW Reference Manual	32108-90001	8/82		32108Q,S,T
HPEASYCHART Reference Manual	32109-90001	8/82*		32109Q,S,T
HPWORD Reference Guide	32120-90001	2/81		32120Q,S,T
HPWORD Quick Reference Guide	32120-90002	2/81		32120Q,S,T
Report/3000 User's Guide	32245-90001	6/82*		32245Q,S,T 32248Q,S,T
HP Inform/3000 User's Guide	32246-90001	4/82		32246Q,S,T
IDS/CHAR Reference Manual	36581-90001	8/81		36581Q,S,T
IDS/FORM Reference Manual	36581-90002	8/81		36581Q,S,T
IFS/3000 Reference Manual	36580-90001	8/81	8/82	36580Q,S,T
TDP/3000 Ref. Man.	36578-90001	11/81		36578Q,S,T
Using TDP/3000	36578-90002	1/80		36578Q,S,T
TDP/3000 Quick Reference Guide	36578-90003	1/80		36578Q,S,T

MANUFACTURING APPLICATIONS MANUALS

Manual Title	Part Number	Edition Date	Updated	Subscription Service
Master Production Scheduling and Rough Cut Resource Planning	32260-90001	7/80		32260Q,S,T
Maintaining Parts and Bills of Material	32260-90002	7/80		32260Q,S,T
Maintaining Routings and Workcenters	32260-90003	7/80		32260Q,S,T
Material Issues and Receipts	32260-90004	7/80		32260Q,S,T
Maintaining Work Orders	32260-90005	7/80		32260Q,S,T
Managing Inventory Balances	32260-90006	7/80		32260Q,S,T
Maintaining Purchase Orders	32260-90007	7/80		32260Q,S,T
Material Requirements Planning	32260-90008	7/80		32260Q,S,T
Standard Product Costing	32260-90009	7/80	2/81	32260Q,S,T
System Customization	32260-90010	7/80	2/81	32260Q,S,T
System Operation	32260-90011	7/80		32260Q,S,T
Materials Mgt/3000 Manual Set	32263A	7/80		
Production Mgt/3000 Manual Set (including keyboard overlays)	32270Z	7/81		
Set of ten keyboard overlays (five for HP3075 & five for HP3076)	32270-60003	7/81		
Defining the Shop	32270-90001	7/81		322270Q,S,T
Managing the Order	32270-90002	7/81		322270Q,S,T

MANUFACTURING APPLICATIONS MANUALS (Continued)

Manual Title	Part Number	Edition Date	Up-dated	Subscription Service
Capacity Requirements Planning and Input/Output Analysis	32270-90003	7/81		322270Q,S,T
A User`s Guide to the Data Capture Terminal	32270-90004	7/81		322270Q,S,T
System Customization	32270-90005	7/81		322270Q,S,T
System Operation	32270-90006	7/81		322270Q,S,T
EDC/3000 User Ref. Man.	32380-90001	3/78	4/79	
EDC/3000 System Admin. Reference Manual	32380-90002	3/78	4/79	32380Q,S,T
EDC/3000 Programmer's Reference Manual	32380-90003	3/78		32380Q,S,T
IOS/3000 User Ref. Man.	32384-90001	3/78		32384Q,S,T
IOS/3000 System Admin. Reference Manual	32384-90002	3/78		32348Q,S,T
IOS/3000 Programmer's Reference Manual	32384-90003	3/78		32348Q,S,T
MRP/3000 User-Admin. Reference Manual	32388-90001	8/78	11/79	32388Q,S,T
MRP/3000 Programmer's Reference Manual	32388-90002	9/78		32388Q,S,T
SPC/3000 User Ref. Man.	32392-90001	4/79		32392Q,S,T

ADDITIONAL MANUALS

		Edition	Up-	Subscription
Manual Title	Part Number	Date	dated	Service
HP 3000 Series System Support Log	03000-90117	2/80	7/82	
HP 3000 Computer System Site Planning Set (Encompasses the 2 manuals below)	30000-60029	6/80	12/81	
HP 3000 Computer System Site Planning and Preparation Guide	30000-90206	6/80	12/80	
HP 3000 Computer System Site Planning Wkb	30000-90207	6/80	12/81	
HP 3000 Computer System Machine Instruction Set	30000-90022	2/80		
Technical Writer's Survival Kit	30000-90171	7/79		
Index to MPE Reference Documents	30000-90045	4/81		FOS
Series 30/33 Diagnostic Manual Set	30070-60068	1/81	3/82	
Series 33 Installation Manual	30070-90021	1/81	6/82	
Series 30 Installation Manual	30080-90001	1/80	7/82	
IBM System/3 to HP 3000 Conversion Guide	32104-90004	7/78		

TRANSACTION PROCESSING MANUALS

Manual Title	Part Number	Edition Date	Up-dated	Subscrition Service
QUERY Reference Manual	30000-90042	1/81		FOS, 30095Q,S,T
KSAM Reference Manual	30000-90079	5/79	5/81	FOS, 32208Q,S,T
HP VPLUS/3000 Ref. Man.	32209-90001	2/81*		FOS, 32209Q,S,T
HP V/3000 Entry Program	32209-90003	1/80		FOS, 32209Q,S,T
Using HP V/3000	32209-90004	1/80		FOS, 32209Q,S,T
IMAGE Data Base Management Reference Manual	32215-90003	9/79	8/81	FOS, 30095Q,S,T
DICTIONARY/3000 Reference Manual	32244-90001	12/81*		32244Q,S,T 32248Q,S,T
TRANSACT/3000 Reference	32247-90001	12/81*		32247Q,S,T 32248Q,S,T

OPT/3000 Enhancements

by Danny Ku, Computer Systems Division

Editor's Note: This article is printed on perforated pages so that you may remove it and insert it in your OPT/3000 Reference Manual. It should be used as a temporary supplement, as all information will be included in the next edition of the manual.

Introduction

This document lists all the enhancements that have been added to the A.00.10 version of OPT/3000. The enhancements are the result of last July's user survey.

Changes for HP 3000 Series 64 Support

Several changes have been made to support the increased CPU power and additional memory of the Series 64:

o If OPT is run on a Series 64, a rate bar of 0-100 will automatically be used in all rate displays, otherwise the original 0-50 rate bar is used. For all systems, the numeric rate is displayed to the right of the rate bar. (This is different from the original implementation, which displayed the numeric rate only when the rate exceeded 50.)

o If the system has more than 2 Mb of main memory, it is impossible to display the entire contents of memory on a single screen of the Memory Contents Display in the Memory Context. Instead, 2 Mb of memory contents are displayed per screen; pressing <RETURN> will generate the display for the next 2 Mb of memory. At end of memory, the next <RETURN> will reset the display to the first 2 Mb of memory. A "K" command will maintain the display at the current 2 Mb of memory; subsequent <RETURN>s will update that block only. A second "K" will again allow the display to cycle through successive blocks of memory with each <RETURN>.

The PARM and INFO Parameters of the :RUN OPT Command

By using the PARM parameter, one can specify an initial context rather than the default Global context. Here is a list of PARM values and their corresponding contexts:

PARM = 0	Global	context (default)
1	Memory	context
2	CPU/MM	context
3	I/O	context

4	Process context
5	Tables context

If the PARM parameter is used, the selected context will be accessed without prompting the user. If an invalid PARM value is entered, the default context (Global) will be used.

A second way to initially specify a context is to use the INFO parameter. The string passed from the INFO parameter is used as an XEQ sequence. This allows users to specify custom command sequences, either interactively or as UDCs. Note that the INFO string must start with a "#" or it will be ignored.

Display Control Commands

Display Control commands access a set of auxiliary operations and are available from any display. Several commands have been changed:

o XEQ (X)

1. The user can now enter an XEQ sequence via the INFO parameter mentioned above.

2. The XEQ sequence will now accept up to 70 characters, compared with 50 characters before.

3. While an XEQ sequence is in effect, the user can enter any display control command without destroying the XEQ sequence. For example, if the user has an XEQ sequence active, he can enter "T" to change the sampling interval. After the interval has been changed, the next command in the XEQ sequence will be executed.

   A useful feature is that the user can now terminate OPT directly from an XEQ sequence by entering an "E" command interactively. The message

   ```
   CONTINUE EXECUTION?  (YES/NO)
   ```

   will be displayed and an "N" will terminate OPT.

4. If an XEQ sequence is in effect and OPT is expecting an input from the user, it will first look for the answer in the XEQ sequence. If the answer is not found, then a <RETURN> by the user is assumed and the display is updated accordingly. The only exceptions are the Program File Users Display and Process Display in the Process context and the Device Display in the I/O Context, where the user will be prompted if no valid input is included in the XEQ sequence. This will allow minimum user intervention in an XEQ sequence. The details will be discussed in the appropriate contexts and displays.

- o Time change (T)

  1. The message prompting the user for a time change now includes the current time interval as well as the maximum allowable. The new prompt is:

     ```
     ENTER THE DELAY BETWEEN DISPLAYS? (CURRENT= xx,MAX= 900 SECONDS)
     ```

     A <RETURN> indicates that no change is required to the interval.

  2. If "T" is included in an XEQ sequence, the user will not be prompted for the time change. Instead, the characters following "T" will be examined to determine the time change. If no number is included, then no change will be made.

- o Listing (L)

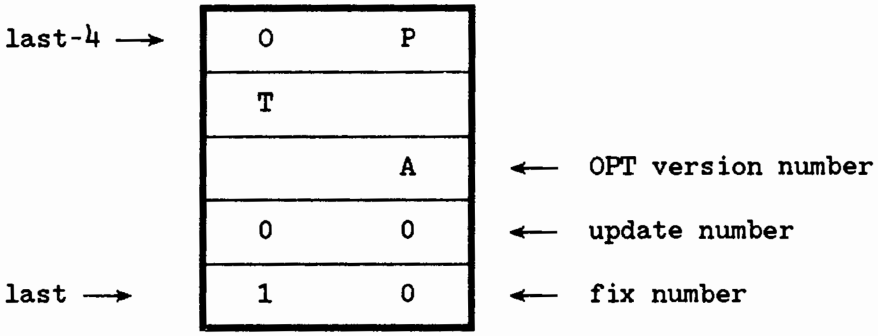
  The user-specified label has been increased from 25 to 50 characters.

OPT Log File

Several changes have been made to the log file:

- o When OPT acquires a log file, it first closes it as a permanent file and then reopens it. This ensures the log file will not be lost if OPT is aborted.

- o New information has been added to the log file. The OPT version, update, and fix numbers are included in the last five words of the header record. The following example shows OPT/3000 A.00.10:



last-4 →	O	P	
	T		
		A	← OPT version number
	0	0	← update number
last →	1	0	← fix number

- o The third word (word 2) of each summary record contains the number of jobs and sessions running at time of the sample. The left byte is the number of jobs, and the right byte is the number of sessions, in binary.

Context Display Changes

Following is a brief description of the changes made to various context displays:

o Global (#G)

In addition to the rate bar changes discussed above, the Global Display
o Summary Report Identification has been increased from 25 to 50 characters.
CPU/Memory Manager (#C)

In the CPU Usage Display, Mean-Time-in-State is shown in milliseconds instead of seconds, and the MAXIMUM OBSERVED STATE PERCENTAGES are replaced by the current and overall CPU state percentages. Also included in the display are the current and maximum "average short transaction times".

o Memory (#M)

Some memory-resident MPE segments are now located in memory banks other than Bank 0. A new segment identifier "R" is used to represent these areas. This change appears in the Memory Contents and Bank Contents displays.

In systems with more than 2 Mb main memory, the display is limited to 2 Mb per screen, as discussed in "HP 3000 Series 64 Support" above.

The free area in the Bank Contents display consists of available spaces, holes, or overlay candidates. OPT counts each of these subregions as a free area, but all neighboring free areas are represented as a single continuous "FREE AREA". As a result, the count of the free areas on the right of the Bank Contents display may not agree with the number of holes in the video display.

If the Bank Contents Display command "B" appears in an XEQ sequence, and the "B" is not immediately followed by a valid octal bank number to be examined, bank 0 will be displayed. OPT will never break from the XEQ sequence to prompt the user for a bank number.

o I/O context (#I)

In the Disc Display, the queue length distribution for disc devices will be displayed if: 1) only one disc device is configured in the system, or 2) the queue length info flag is set by a "Q" command. The default for this flag is True; upon entering disc display, the queue length distribution will be shown along with the DISC I/O rate. Entering a second "Q" will toggle the flag to False.

For device displays, it is no longer necessary to enter exactly four devices. Press <RETURN> or enter "@" to indicate all devices should be shown, four devices at a time. "N" causes the display of the next four devices, while a <RETURN> will merely updates the current devices. Another way to specify the devices desired is to use the range indicator "/"; use "*xx/yy*" to specify all device indexes between *xx* and *yy* (inclusive). If the range is out of bounds, an error message will be printed and the user will be prompted again for devices.

The device map in the device display is no longer used. Instead, logical device numbers are shown.

If queue length distributions are selected, only one disc device's queue length distribution will be shown. An "N" command will move to the next disc display.

If an XEQ sequence requesting a device display contains an "@" or a " " (blank) following the device display command and more than four devices are configured on the the system, then all devices will be displayed. Otherwise, the user will prompted for the selection of devices.

o Process (#P)

1. A User Summary display has been added to this context. The following information is displayed horizontally across the screen for each PIN:
   PIN number
   User and account names
   Fully qualified program name
   Cumulative CPU time (in milliseconds)
   CPU percentage time during last time interval
   Priority number of the process
   Stack size
   Total code segment size *
   Total extra data segment size *

   * These values are for segments in the process's working set and are in memory only.

   The first time the User Summary Display appears, the display reports user processes only. An "A" will display all processes, including system and CI processes; a second "A" will return to the user process display. Any process displayed for the first time will appear in inverse video.

   In general, if the user has SM or OP capability, or has AM for the logon account of the process, then the program file name (for a user process) or the last CI command (for a CI process) will also be displayed.

   The CPU column reports milliseconds of CPU time used. If the figure exceeds 999999 ms, then it is converted to seconds and an "S" will be appended. If more than 9999 seconds are used, then the CPU column reports minutes, and an "M" is appended.

   Entering a "P" in the User Summary Display will go to the Process Display directly without stepping through the Program File and Program File Users Displays. The user is prompted with:

   ```
   ENTER PINS OF PROCESSES FOR WHICH INFORMATION IS DESIRED
      ('@' OR <CR> FOR ALL)
   ```

   There are several ways to select the PINs. A single PIN number or a list of PINs separated by commas (,) may be used. A range "*xx/yy*" will indicate that PINs in the display list are to be between *xx* to *yy*, inclusive. An "@" or a <RETURN> places all PINs into the display list.

As usual, if more than one process is requested, then an "N" command will generate the Next Process display. At the end of the list, OPT will go back to the previous display, either the User Summary or Program File Users.

2. The Process Display now also includes the capability list of the user. The list is in the box where cumulative CPU time and the process priority number are displayed. The list is arranged in three groups. Only those capabilities actually possessed by the user are displayed on each line:

   Group 1: SM, AM, OP, AL, GL
   Group 2: DI, UV, LG, ND, SF, CS
   Group 3: PM, PH, DS, MR, IA, BA

   Another feature added to the Process Display is the ability to display the user's stack markers. This is activated with an "M" command. If a new process is added to the display, the "M" command must be repeated. The "M" command will remain enabled if the displayed process is not changed.

   If the stack marker is altered during the period that the sampling is changing from one marker to another, a line of asterisks (*****) will appear on the display if the marker is determined to be invalid. If all markers are displayed and terminated with the segment names "MORGUE" or "MOR'RIN'ABDP", the stack marker display is valid.

   The following information appears in the Stack Marker Display:

   o The stack-relative address of delta Q.

   o The Q-7 to delta Q values. If delta Q is less than 8, then only those values up to the previous stack marker are displayed (except the very last stack marker's values, for which all 8 words are shown).

   o The segment name (for user segment, no name is given.)

3. An "S" command will list all the locked SIRs in the system. The list includes the SIR number, its name, a holder PIN and a list of all PINs that were impeded for that SIR. If no SIRs are locked, then the message "(no locked sirs)" will be displayed instead.

4. If an XEQ sequence is in effect, then an "@" or a " " (blank) following a "U" (Program File Users Display) or a "P" (Process Display) will indicate "display all", and no user prompt will be given. Any other character will cause the user to be prompted for the list.

5. The INUSE flag of the Program Files User Display is set to True by default.

o Tables (#T)

Seven new tables have been included. The tables have been ordered so that the displays' orders will be as close to that of SYSDUMP as possible. Because of space limitations, the tables are presented in two screens. A <RETURN> will change the display to the next screen, and a "K" command will keep the display at the current screen until another "K" command is entered.

The list of all displayed tables follows; an asterisk indicates a new entry:

First screen:

CODE SEGMENT
EXTENDED CODE SEGMENT
DATA SEGMENT
PROCESS
I/O QUEUE
DISC REQUEST
TERMINAL BUFFERS
ATP TERMINAL BUFFERS(*)
SYSTEM BUFFERS
SWAP TABLES
CST BLOCK

Second screen:

PRIMARY MESSAGE(*)
SECONDARY MESSAGE(*)
SPECIAL REQUEST(*)
INTERRUPT CONTROL STACK
UCOP REQUEST(*)
TIMER REQUEST
BREAK POINT(*)
RIN TABLE(*)
JOB PROCESS COUNT
VIRTUAL MEMORY
SPOOLER DISC SPACE

For hard copy listings, all tables will be listed even though only one set of tables is displayed on the screen at a time.

List of OPT/3000 Commands

Default Displays and values are underlined.

```
DESCRIPTION:                                              APPLICABLE ADDITIONAL
                                                                      COMMANDS:


#C - CPU/MEMORY MANAGER CONTEXT COMMANDS (PARM=2)

   Displays: ?  CPU Context Menu
             C  CPU Usage
             M  Memory Management Activity
             R  CPU-Memory Management Report


#G - GLOBAL CONTEXT COMMANDS (PARM=0)

   Displays: ?  Global Context Menu
             G  Global Resource Usage Report                    A S R Z
             P  Process Summary                                 A S R Z

  Additional A  Activate summary report generation
   Commands: R  Generate single summary report
             S  Stop summary report generation
             Z  Zero totals


#I - I/O CONTEXT COMMANDS (PARM=3)

   Displays: ?  I/O Context Menu
             D  Disc Activity                                   A N Q
             M  Magnetic Tape Activity                          A N
             P  Line Printer Activity                           A N
             R  I/O Activity Report

  Additional A  Alter displayed devices
   Commands: N  Next
             Q  Queue length information (True)
```

#M - MEMORY CONTEXT COMMANDS (PARM=1)

Displays:	?	Memory Context Menu	
	B	Bank Contents	
	C	Code Segment Size Histogram	A N P U
	D	Data Segment Size Histogram	A N P U
	F	Free Area Size Histogram	
	M	Memory Contents	K
	O	Overall Histogram	A N P U
	R	Memory Report	
	S	Stack Segment Size Histogram	A N P U
Additional	A	All segments	
Commands:	K	Keep current memory display	
	N	Non-user segments only	
	P	Present segments only	
	U	User segments only	

#P PROCESS CONTEXT COMMANDS (PARM=4)

Displays:	?	Process Context Menu	
	F	Program File	I S
	G	User Summary	S
	P	Process	M N S W
	R	Process State Report	
	U	Program File Users	N S W
Additional	A	Display all proceses (False)	
Commands:	I	In use only (True)	
	M	Stack markers (False)	
	N	Next	
	S	Display locked SIRs.	
	W	List working set segments (False)	

#T TABLES CONTEXT COMMANDS (PARM=5)

Displays:	?	Tables Context Menu	
	D	Detailed Table Information	C K
	R	Table Utilization Report	C K
Additional	C	Clear observed Maximums	
Commands:	K	Keep current display (False)	

SEGMENTER Enhancements

by Jessy J. Hsu, Computer System Division

Editor's Note: The information in this article was finalized after the new (November 1982) edition of the MPE Segmenter Reference Manual (30000-90011) was printed. This article is printed on perforated pages so that you may remove it and insert it in your Segmenter manual. This information will be incorporated in the next update to the manual.

Most experienced users of MPE are familiar with the PMAP option of the Segmenter -PREPARE and -ADDSL commands and the system :PREP command. These cause the Segmenter to produce a listing of the locations of procedures and their entry points, plus external references, in all segments prepared.

The PMAP information most useful to the user is that which identifies the locations of procedures and their entry points within a given code segment. We have now added a mechanism to store an internal copy of most of the information contained in the PMAP as part of a program or SL file; this is called the FPMAP information. External references to procedures in other segments are not saved as part of the program and SL file FPMAP information. A new command and new user-callable procedures are provided to access the FPMAP information.

This information is especially helpful for users of APS/3000 and HPToolset. The inclusion of the FPMAP information as part of a program/SL file provides a mapping of segment-relative addresses back to the source level procedure names and entry points. Application Program SAMPLER (APS/3000), for instance, uses this information to provide detailed execution profiles of any monitored program, in terms of procedure names and code within each procedure. This information can be used to quickly determine the CPU bottleneck of almost any program on a HP/3000 system. When the System FPMAP flag is set, the FPMAP data is automatically included in each program file at PREP time. Thus, the procedure execution profile of any active program in the installation can be readily obtained by the installation manager or development engineers.

Performance Considerations

The program file size increases about 3 to 10 percent if the FPMAP information is included in the file. CPU time for preparing a program increases about 5 to 20 percent when a program is prepared with FPMAP.

Controlling the Inclusion of FPMAP Information

Several new features have been added to the Segmenter which allow system managers and users to control the FPMAP inclusion in program and SL files. The inclusion of the FPMAP information depends on the conditions of the System-wide FPMAP option flag and the Job/Session-wide FPMAP option flag, and the new FPMAP and NOFPMAP parameters of the :PREPARE, -PREPARE, and -ADDSL commands; figure 1 illustrates the interaction.

Notice that the Command parameters override the Job/Session FPMAP flag and the Job/Session FPMAP flag overrides the System FPMAP flag, unless the System FPMAP flag is set unconditionally. The System FPMAP flag can be set to conditional or unconditional. If it is set to unconditional, all users on the system will have FPMAP information included in their programs and SL files when they prepare programs or add segments to SL files, regardless of the Job/Session FPMAP flag and parameters specified. If the System FPMAP flag is conditional, then it can be overridden by lower levels of FPMAP option control. For example, when either the System conditional FPMAP or the Job/Session FPMAP is on, the user can enter the NOFPMAP parameter. This will result in a program without FPMAP information. The System FPMAP flag is initialized to OFF when a system is started or loaded. When a job or a session is logged on, the Job/Session FPMAP flag is initialized to the System FPMAP flag value.

System FPMAP Flag	Job/Session FPMAP Flag	FPMAP/NOFPMAP Parameter	FPMAP Included?
Uncond. On	On or Off	Either	Yes
Cond. On	On	Either	Yes
		FPMAP	Yes
		NOFPMAP	No
	Off	Either	No
		FPMAP	Yes
		NOFPMAP	No
Off	On	Either	Yes
		FPMAP	Yes
		NOFPMAP	No
	Off	Either	No
		FPMAP	Yes
		NOFPMAP	No

Figure 1. Interactions in FPMAP Inclusion

For users of the HPToolset subsystem, the FPMAP will always be included if symbolic debugging is requested, regardless of the flags specified. Only the NOSYM parameter can suppress the FPMAP information in this instance. (See Communicator number 29, "New PREP Command Parameters".)

The new facilities for FPMAP inclusion are:

o A new optional parameter pair has been added to the -PREPARE, :PREPARE, and -ADDSL commands; the default conditions can be determined from figure 1. The syntax is:

```
-PREPARE progname [;map_opt]
-PREPARE progname [;map_opt]
-ADDSL segname [;map_opt]
```

where

```
                       {FPMAP   }
map_opt is defined as  {NOFPMAP}
```

o The -SETFPMAP command has been added to the Segmenter to set the system or job/session FPMAP flag. The syntax is:

```
                            {CONDITION   }
                {SYSTEM  [;{UNCONDITION}]        }
-SETFPMAP   [                                      ][;{ON }]
                {SESSION                          }   {OFF}
```

where

SYSTEM/SESSION — Specifies which flag is to be set/reset. Default is SESSION.
Requires SM capability to set/reset System FPMAP flag

CONDITION/UNCONDITION — Specifies that the System FPMAP flag is set to conditional or unconditional.
Valid only when SYSTEM is specified.
Default is CONDITION.

ON/OFF — Set/Reset flag.
Default is ON.

o The -SHOW command has been added to the Segmenter to display the conditions of the System and Job/Session FPMAP flags. In addition, it lists files currently opened by the Segmenter. The syntax is:

```
-SHOW
```

An example output is

```
-SHOW

USL FILE       :  MYUSL.MYGROUP.MYACCT
AUX USL FILE   :  NONE
SL FILE        :  MYSL.MYGROUP.MYACCT
RL FILE        :  NONE
SYSTEM FPMAP   :  ON (CONDITION)
SESSION FPMAP  :  OFF
```

Accessing the FPMAP Information Interactively

FPMAP information may be accessed interactively through the new -LISTPMAP Segmenter command. The command's syntax is:

```
                      {segname      }
-LISTPMAP progname [;{procedurename}]
```

where

progname Is the name of the program whose PMAP is to be listed.

segname Is the name of the segment whose PMAP is to be listed.

procedurename Is the name of the procedure whose PMAP is to be listed.

The default is to list all segments in the specified program file.

An example output follows:

```
-LISTPMAP MYPROG

PROGRAM FILE MYPROG.MYGROUP.MYACCT

MAIN                 0        4056

  NAME               TYPE     CODE     ENTRY     LENGTH
  PROC1               P          0        10         45
  PROC11              SP                  21
  PROC2               P         45        45       3000
  PROC3               P       3045      3050       1000
```

Obtaining FPMAP Information Programmatically

A new set of user-callable procedures has been added to allow programmatic access to FPMAP information. They must be declared explicitly within the calling program. Declarations appear in the description of each procedure.

In order to provide a unified means of returning FPMAP data, each of these procedures makes use of a special data structure called an external FPMAP record, whose format is given in figure 2.

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Field Defined for
0	Type (0=Segment, 1=Procedure, 2=Sec. Entry)	0, 1, 2
1 8	Segment Name	0, 1, 2
9 16	Procedure Name	1, 2
17 24	Secondary Entry Point Name	2
25	Logical Segment Number	0, 1, 2
26	Segment Length (Including STT)	0, 1, 2
27	STT Length	0, 1, 2
28	Procedure Start Address	1, 2
29	Procedure Length	1, 2
30	Primary Entry Point Address	1, 2
31	Secondary Entry Point Address	2
32	(Not Used)	
33	HPToolset Procedure ID	1
34 35	HPToolset Link	1

Note: All undefined fields will contain zeros or blanks as appropriate.

Figure 2. External FPMAP Record Format

o **FindPmapName**

Declarations

```
procedure FindPmapName (ProgFnum, SegName, EntName, XpmapRec, XpmapRecLen,
                        Status);
   value ProgFnum, XpmapRecLen;
   integer ProgFnum;
   byte array SegName;
   byte array EntName;
   integer array XpmapRec;
   integer XpmapRecLen, Status;

   option variable;
```

Description

This procedure searches the FPMAP in a program or SL file for a segment and/or entry point name. If found, an appropriate external FPMAP record will be returned. Otherwise, the status code will indicate why the search failed.

One or both of the parameters "SegName" and "EntName" must be included in the actual parameter list; the action taken according to their various combinations is as follows:

> "SegName" only: This calling sequence results in a search for a segment name only. If found, a segment (type 0) external FPMAP record will be returned.
>
> "EntName" only: This calling sequence results in a search for a procedure or secondary entry point name. In program files, all procedures and secondary entry points qualify for the search. In SL files, only those procedures and secondary entry points appearing in the file directory will be candidates for a match. This implies that hidden procedures and secondary entry points will not be found. Only a procedure (type 1) or entry point (type 2) external FPMAP record will be returned.
>
> Both "SegName" and "EntName": This calling sequence results in a search for a procedure or secondary entry point name in a specified segment. Hidden procedures and secondary entry points will be included in the search since the segment name qualifies such names for uniqueness.

"SegName" and "EntName" may be excluded from a procedure call either by physical omission in the parameter list, or by the presence of a blank as the first character in their byte arrays. All other parameters in the calling sequence must be present.

Input Variables:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be searched. The caller is responsible for opening the file with MULTIRECORD, NOBUF, and READ access.

SegName - This is the name of the segment to be found. It should be terminated by a blank, but in no case will more than the first 15 characters be accessed for the search. Comparisons will be made with all characters upshifted, but the string passed in "SegName" will be returned unchanged. This is an optional parameter, and interacts with "EntName". Refer to the discussion above for details.

EntName - This is the name of the procedure or secondary entry point to be found. The format restrictions given for "SegName" also apply to this variable. This is an optional parameter, and interacts with "SegName". Refer to the discussion above for details.

XpmapRecLen - This is the number of words of the external FPMAP record the caller wants returned in "XpmapRec". Should the caller request more words than are currently defined for an external FPMAP record, zeroes will be returned in the extra words.

Output Variables:

XpmapRec - This is the external FPMAP record corresponding to the entry point name passed in "EntName". The format of this record has been previously defined in this document. Any of the external FPMAP record types may be returned.

Status - This is a status word containing information about the success or failure of the procedure call as follows:
 0 - No errors.
 1 - The entry point name was not found.
 3 - The segment name was not found.
 9 - An invalid parameter combination was passed.
10 - The file "ProgFnum" did not contain FPMAP data.
11 - The file "ProgFnum" was not a program or SL file.
12 - A file system I/O error occurred on the file "ProgFnum".

Condition Code:

CCE - The symbolic name was found (Status = 0).

CCG - The symbolic name was not found (Status = 1 or 3).

CCL - Unusual error (Status = 9 through 12).

o **FindPmapAddr**

Declarations

```
procedure FindPmapAddr(ProgFnum, SegNum, Address, XpmapRec, XpmapRecLen
                       Status);
   value ProgFnum, SegNum, Address, XpmapRecLen;
   integer ProgFnum, SegNum, Address;
   integer array XpmapRec;
   integer XpmapRecLen, Status;
```

Description

This procedure searches the FPMAP in a program or SL file for a specific address in a particular segment. An external FPMAP record corresponding to the first entry point at or preceding "Address" is returned.

Input Variables:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be searched. The caller is responsible for opening the file with MULTIrecord NOBUF READ access.

SegNum - This is the number of the logical segment in which the address is to be found.

Address - This is the PB-relative address in segment "SegNum" with which the external FPMAP record to be returned is associated.

XpmapRecLen - This is the number of words of the external FPMAP record the caller wants returned in "XpmapRec". Should the caller request more words than are currently defined for an external FPMAP record, zeroes will be returned in the extra words.

Output Variables:

XpmapRec - This is the external FPMAP record corresponding to the first entry point at or preceding "Address". The format of this record has been previously defined in this document. It may be either a procedure (type 1) or secondary entry point (type 2) record, depending on "Address" and whether or not secondary entry points exist. If the located entry point has more than one name, the first one encountered in the FPMAP will be returned.

Status - This is a status word containing information about the success or failure of the procedure call as follows:
  0 - The address was valid and an external FPMAP record was returned.
  2 - The address was out of range.
  3 - Logical segment "SegNum" didn't exist.
 10 - The file "ProgFnum" did not contain FPMAP data.
 11 - The file "ProgFnum" was not a program or SL file.
 12 - A file system I/O error occurred on file "ProgFnum".

Condition Code:

CCE - The address was valid (Status = 0).

CCG - The address or segment number was invalid (Status = 2 or 3).

CCL - File error (Status = 10 through 12).

- **DumpPmap**

Declarations

```
procedure  DumpPmap  (ProgFnum,  PmapFnum,  XpmapRecLen,  RecCount,  SegNum,
                      Status);

   value ProgFnum, PmapFnum, XpmapRecLen, SegNum;
   integer ProgFnum, PmapFnum, XpmapRecLen, RecCount, SegNum, Status;
```

Description

This procedure fills a file with external FPMAP records representing either all of the FPMAP data stored in a program or SL file, or only that for a specified segment. One external record will be created for, and in the same order as, each internal FPMAP record encountered.

Note that if multiple names exist for the same secondary entry point address, one FPMAP record will be created for each of the names. "DumpPmap" is the only procedure which provides access to such multiply defined names.

Input Variables:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be dumped. The caller is responsible for opening the file with multirecord, read-only access.

PmapFnum - This is the number of the file, as returned from the 'Fopen' intrinsic, which is to contain the external FPMAP records. The caller is responsible for opening the file with write access and fixed-length records of size at least "XpmapRecLen" words.

XpmapRecLen - This is the number of words in each external FPMAP record to be written to file "PmapFnum". Should the caller request more words than are currently defined for an external FPMAP record, the extra words in each record written will be set to zero.

SegNum - This is the number of the segment whose FPMAP data are to be dumped. A value of -1 causes all FPMAP records for all segments in the file to be dumped.

Output Variables:

RecCount - This is the number of external FPMAP records which were, or could have been, generated from the FPMAP data found in the program or SL file. Assuming the caller's PmapFnum file was large enough, this is also the number of records written to that file. Status - This is a status word containing information about the success or failure of the procedure call as follows:

 0 - The external FPMAP file was built successfully.
 4 - The external FPMAP file was not large enough.
10 - The file "ProgFnum" did not contain FPMAP data.
11 - The file "ProgFnum" was not a program or SL file.
12 - A file system I/O error occurred on the file "ProgFnum".
13 - A file system I/O error occurred on file "PmapFnum".

Condition Code:

CCE - The FPMAP file was built successfully (Status = 0).

CCG - The FPMAP file was not large enough (Status = 4).

CCL - File error (Status = 10 through 13).