HP 3000 Computer Systems

MPE V System Utilities Manual

HP Part No. 32033-90008
Printed in U.S.A. 1989

Third Edition
E1089

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

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

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

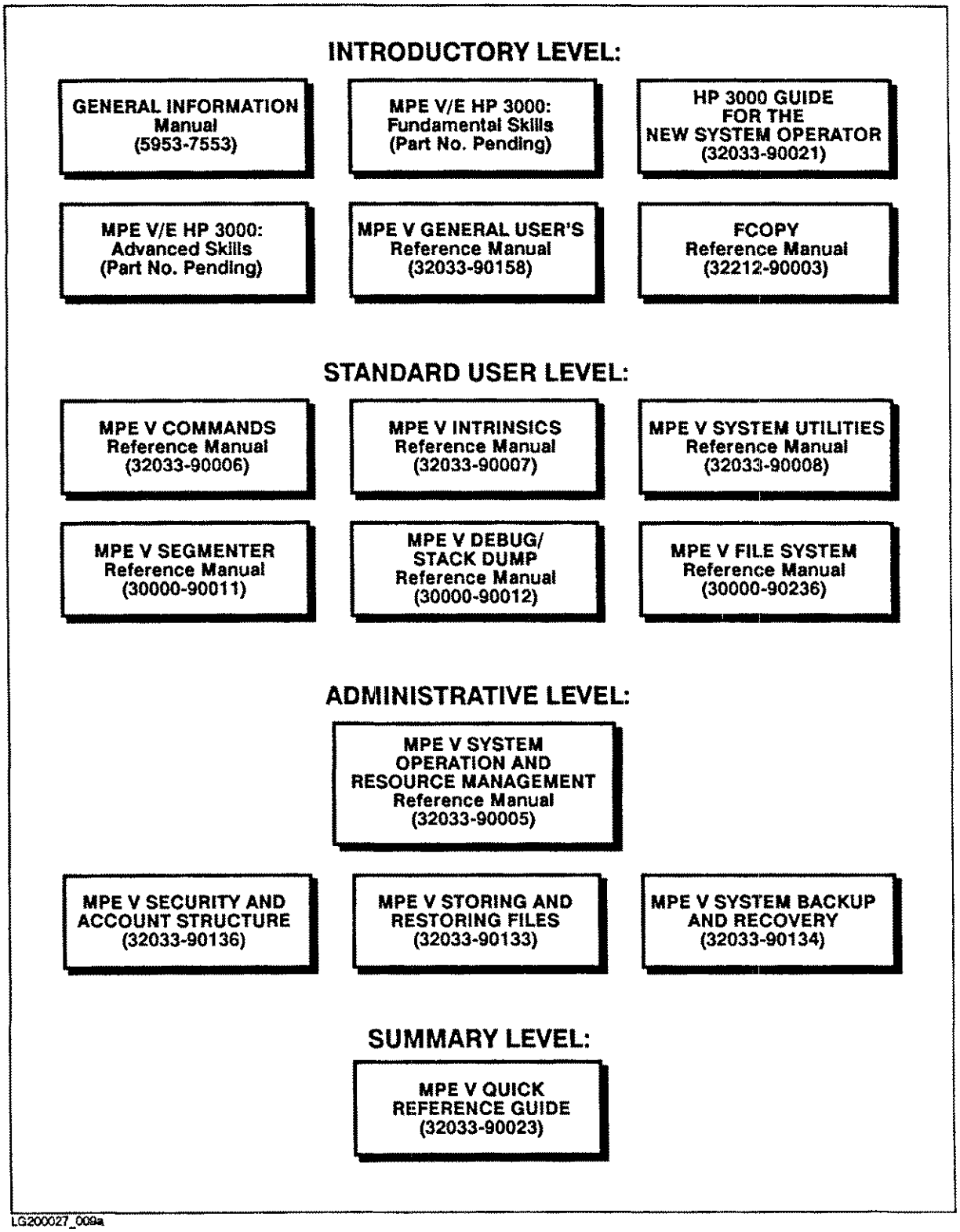
Copyright © 1989 by Hewlett-Packard Company

Print History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	Date	Software Version
First Edition	Jul 1984	G.00.00
Update #1	Jan 1985	G.01.00
Second Edition	Feb 1986	G.02.00
Update #1	Aug 1986	G.02.B0
Update #2	Jun 1987	G.03.00
Update #3	Oct 1988	G.03.04
Third Edition	Oct 1989	G.03.08

MPE V Manual Plan



LG200027_009a

Preface

This manual documents the utility programs that are part of the multi-programming executive (MPE) operating system running on the HP 3000 computer.

The manual is designed for users at many skill levels, including general users who can log on to the system and want information on how much free disk space is available, system managers who wants to know how system resources are being allocated, and programmers who are making online changes to program code. The capability requirements for using each program are an indication, but not an infallible guide, as to the complexity of the program and the extent of its impact on the system.

How to Use This Manual

This manual contains 16 chapters and an appendix. Chapter 1 introduces the utilities covered in the next 15 chapters. Chapters 2 through 16 describe the utilities in detail and give the following information about them:

- Syntax
- Operation
- Commands (if any)
- Examples
- Error messages

Appendix A lists abbreviations for user capability sets and access modes.

Additional Information

If you need help or more information, the following supporting documentation provides in-depth discussions of concepts used by the utility programs:

- *MPE V Intrinsics Reference Manual* (32033-90007)
- *MPE System Operation and Resource Management Reference Manual* (32033-90005)
- *MPE V/E Commands Reference Manual* (32033-90006)

Conventions

UPPERCASE

In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

```
COMMAND
```

can be entered as any of the following:

```
command        Command        COMMAND
```

It cannot, however, be entered as:

```
comm           com_mand       comamnd
```

italics

In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *filename* with the name of the file:

COMMAND *filename*

bold italics

In a syntax statement, a word in bold italics represents a parameter that you must replace with the actual value. In the following example, you must replace ***filename*** with the name of the file:

COMMAND(***filename***)

punctuation

In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

(*filename*) : (*filename*)

underlining

Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

```
Do you want to continue? >>  yes
```

{ }

In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either ON or OFF:

```
COMMAND { ON  }
        { OFF }
```

Conventions (continued)

[ ]

In a syntax statement, brackets enclose optional elements. In the following example, `OPTION` can be omitted:

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

`COMMAND` *filename* [`OPTION` / *parameter*]

[...]

In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *parameter* zero or more times. Each instance of *parameter* must be preceded by a comma:

[,*parameter*] [...]

In the example below, you only use the comma as a delimiter if *parameter* is repeated; no comma is used before the first occurrence of *parameter:*

[*parameter*] [, ...]

| ... |

In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select `A`, `AB`, `BA`, or `B`. The elements cannot be repeated.

{`A` / `B`} | ... |

...

In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

Δ

In a syntax statement, the space symbol Δ shows a required blank. In the following example, *parameter* and *parameter* must be separated with a blank:

(*parameter*)Δ(*parameter*)

[ ]

The symbol [ ] indicates a key on the keyboard. For example, [RETURN] represents the carriage return key or [Shift] represents the shift key.

[CTRL]*character*

[CTRL]*character* indicates a control character. For example, [CTRL]Y means that you press the control key and the Y key simultaneously.

Conventions (continued)

base prefixes

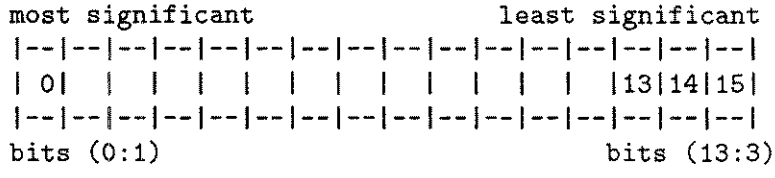
The prefixes %, #, and $ specify the numerical base of the value that follows:

%num specifies an octal number.
#num specifies a decimal number.
$num specifies a hexadecimal number.

If no base is specified, decimal is assumed.

bits (*bit:length*)

When a parameter contains more than one piece of data within its bit field, the different data fields are described in the format bits (*bit:length*), where *bit* is the first bit in the field and *length* is the number of consecutive bits in the field. For example, bits (13:3) indicates bits 13, 14, and 15:



Contents

1. Introduction

User Capabilities 1-1
Utilities Covered in This Manual 1-1
General System Utilities 1-1
Administrative Utilities 1-2
Error Recovery Utilities 1-2
File Modification Utilities 1-2

2. LISTEQ2/LISTEQ5

Operation 2-1
Example 2-2
LISTEQ2/LISTEQ5 Error Messages 2-2

3. LISTDIR2/LISTDIR5

Operation 3-1
Commands 3-2
Security Restrictions Within LISTDIR5 3-2
Generic Names and Wildcard Characters 3-3
Private Volumes 3-4
LISTACCT 3-5
LISTGROUP 3-6
LISTUSER 3-7
LISTF 3-8
LISTSEC 3-10
MOUNT 3-12
DISMOUNT 3-13
EXIT 3-14
HELP 3-15
LISTDIR2/LISTDIR5 Error Messages 3-16

4. SPOOK/SPOOK4/SPOOK5

Operation 4-1
MPE V/P and MPE IV Systems 4-1
MPE V/E Systems 4-1
Multi-Version and Upgraded Sites 4-2
Commands 4-2
Running SPOOK 4-3
Entering SPOOK Commands 4-3
Syntax for a Range List 4-4
Examples of Range Lists 4-4
ALTER 4-5
APPEND 4-7

COPY 4-9
DEBUG 4-11
EXIT 4-12
FIND 4-13
HELP 4-15
INPUT 4-16
KILL 4-18
LIST 4-19
MODE 4-21
OUTPUT 4-22
PURGE 4-24
QUIT 4-25
RUN 4-26
SHOW 4-28
TEXT 4-31
XPLAIN 4-32
SPOOK4/SPOOK5 Messages 4-33

5. FREE2/FREE5
Operation 5-1
Example 5-2
FREE2/FREE5 Error Messages 5-2

6. MEMLOGAN
Operation 6-1
MEMLOGAN Environment 6-2
Memory Error Log Formats 6-2
MEMLOGAN on Series 37 6-2
MEMLOGAN on Series 30/39/40/42/44/48/52/58 6-3
Interactive Input for Series 4X/5X Only 6-3
MEMLOGAN on Series 64/68/70 6-4
Examples 6-5
Series 37 MEMLOGAN Output 6-5
Series 39/40/42/44 /48/52/58 MEMLOGAN Output 6-6
Series 64/68/70 MEMLOGAN Output 6-7
MEMLOGAN Error Messages 6-8

7. MEMTIMER
Operation 7-1
Example 7-2
MEMTIMER Error Messages 7-2

8. LISTLOG2/LISTLOG5
Operation 8-1
Examples 8-3
Auditing User Activity 8-6
LISTLOG2/LISTLOG5 Error Messages 8-8

9. ASOCTABL/ASOCTBL5
Operation 9-2
Examples 9-3
ASOCTABL/ASOCTBL5 Error Messages 9-4

10. DPAN4/DPAN5
Taking a Memory Dump 10-1
Memory Dump on Series 37 10-1
Notes 10-3
Memory Dump on Series 39/40/42/44/48/52/58 10-4
Memory Dump on Series 64/68/70 10-5
DPAN5 Dump Analysis Options 10-6
Producing a DPAN5 Listing Using a Batch Job 10-6
Producing a DPAN5 Listing Interactively 10-6
Running DPAN5 Without the INFO String 10-7
Running DPAN5 With the INFO String 10-7
Examples 10-9
DPAN4/DPAN5 Error Messages 10-11

11. SADUTIL
Preparing a SADUTIL Coldload Medium 11-2
Loading and Running SADUTIL 11-3
Coldloading the DUS on the Series 37 11-3
Coldloading the DUS on Series 39/40/42/44/48/52/58 11-3
Coldloading the DUS on Series 64/68/70 11-4
SADUTIL Configuration Dialog 11-4
SADUTIL Device Types and Subtypes 11-7
CLID 11-8
CONF 11-9
COPY 11-10
EDIT 11-11
FIND 11-13
HELP 11-15
OUTM 11-16
PDSK 11-17
PDTT 11-19
PFIL 11-20
PVOL 11-22
SAVE 11-23
STOP 11-27
SADUTIL Error Messages 11-28

12. RECOVER2/RECOVER5
Operation 12-1
Examples 12-3
RECOVER2/RECOVER5 Error Messages 12-4

13. DISKED2/DISKED5
Operation 13-1
Commands 13-2
BASE 13-3
DEBUG 13-4
DISC 13-5
DUMP 13-6
EXIT 13-7
FILE 13-8
HELP 13-9
LIST 13-10
MODIFY 13-11
WIDTH 13-12
DISKED2/DISKED5 Error Messages 13-18

14. PATCH
Operation 14-1
Using PMAP 14-2
Running PATCH 14-4
D 14-5
M 14-6
DG 14-7
MG 14-9

15. SLPATCH
Running SLPATCH 15-1
D 15-2
M 15-3
SLPATCH Error Messages 15-4

16. VINIT
Operation 16-1
Commands 16-2
>COND 16-2
>COPY 16-2
>DSTAT 16-2
>DTRACK 16-3
>EXIT 16-3
>FOREIGN 16-3
>FORMAT 16-3
>HELP 16-3
>INIT 16-4
>PDEFN 16-4
>PDTRACK 16-4
>PFSPACE 16-4
>PLABEL 16-4

>SCRATCH 16-5
>SERIAL 16-5
>VERIFY 16-5
Command Parameters 16-6

A. Capabilities and Access Modes
Capability Sets A-1
User Attributes A-1
File Attributes A-1
Program Attributes A-2
Access Mode Abbreviations A-2
User-Type Abbreviations A-3

Index

1

Introduction

Utilities are programs of general usefulness, applicable to many tasks and purposes, that perform functions not available through MPE commands. Utilities are used in program development, file manipulation, and system administration to provide information access controls, report on system resources and gather other special-purpose information. Certain utilities operate in a stand-alone mode, that is, without MPE running.

Many of the utilities in this manual have two versions, one for MPE V/E (G.00.00 and later), and one for MPE IV and MPE V/P (E/F.00.00). This is indicated in the chapter titles by dual names, separated by a slash, as in "LISTEQ2/LISTEQ5". In all cases, the dialog and operation of the versions is the same. The only difference is the name of the file in which the program resides. Full details are presented in each chapter.

User Capabilities

The MPE operating system is organized in such a way that each user of the system is assigned a capability set. The capability set can be unique to each user, or to each account, or to a group. However, it always consists of the sum of user attributes, file attributes, and program attributes. Some functions within utility programs are limited by the capability set of the program's user. A brief description of the requirements of each is given below; more information is provided in the individual program descriptions. Appendix A contains an explanation of all of the user, file, and program attributes.

Utilities Covered in This Manual

The MPE V system utilities fall into three main groups. They are listed below.

General System Utilities

The following utilities are used by general system users. Some have special features available to users with nonstandard capabilities.

- LISTEQ2/LISTEQ5 lists the file equations and temporary files for the current session. No special capabilities are required.
- LISTDIR2/LISTDIR5 lists the attributes of accounts, users, groups, and files. All users may list the attributes of their logon account, group, user name, and files (although some attributes are restricted). The account manager may list the attributes of other groups and users within the logon account, including passwords, lockwords, and creators. System managers may list the attributes of any file, group, user, or account in the system.

- SPOOK/SPOOK4/SPOOK5 displays, modifies and moves spooled device files. The extensive command set (18 functions) allows a variety of access levels. General users can access their own files, but may not use the `INPUT` or `OUTPUT` commands. An account manager (AM) may access any spool file on the logon account, but cannot use the `INPUT` or `OUTPUT` commands. The system manager (SM) may access any spool file and use all commands. All users, including account and system managers must have PM capability to use the `DEBUG` command.

Administrative Utilities

System managers and other administrative users may use these utilities:

- FREE2/FREE5 details the contiguous free space and total free space on disks, and the total free space in the system. No special capabilities are needed.
- MEMLOGAN prints out error logs recorded by the memory error logging system from the memory error logging boards. System manager (SM) capability is required.
- MEMTIMER sets the update intervals of the memory logging file used by MEMLOGAN. System manager (SM) capability is required.
- LISTLOG2/LISTLOG5 analyzes MPE system log files. It allows you to print events from a particular log file or files. SM capability is required.
- ASOCTABL/ASOCTBL5 allows a system manager to define which users are authorized to control which device classes. This utility can also be used to gain exclusive access to a device.

Error Recovery Utilities

The following utilities are used to recover from and analyze system problems:

- DPAN4/DPAN5 produces a formatted dump listing of the main memory and virtual storage from the dump taken by the software dump facility. No special capabilities are required to run it.
- SADUTIL performs emergency disk operations after the system has gone down. This is a stand-alone program with no operating system controls. No special capability is required, but SM or system supervisor (OP) capability is recommended.
- RECOVER2/RECOVER5 reloads the files created by SADUTIL back into the system after a system failure. System manager capability is required. This utility runs only in privileged mode and within sessions.

File Modification Utilities

Finally, the following utilities are often used by applications programmers and system managers to modify disk files:

- DISKED2/DISKED5 allows you to display and/or modify the sector contents of a disk. This utility runs in privileged mode and within sessions.
- PATCH displays and modifies the object code of programs without recompiling. No special MPE capabilities are required to run PATCH.
- SLPATCH allows you to display or modify the contents of a segmented library file. SLPATCH runs in privileged mode, but does not require special capabilities.

2

LISTEQ2/LISTEQ5

LISTEQ2/LISTEQ5 lists the file equations and temporary files in existence for the current session. No special MPE capabilities are required to run this program.

The program LISTEQ5.PUB.SYS is found only on MPE V/E; the MPE V/P (E/F.00.00) version is LISTEQ2.PUB.SYS. The dialogue and operation of the two are identical. Throughout this chapter, LISTEQ5 is used for all examples and explanations.

Operation

The following steps describe the operation of LISTEQ5. Enter the commands at the MPE colon prompt (:).

1. To run LISTEQ5, type:

```
:RUN LISTEQ5.PUB.SYS
```

2. LISTEQ5 identifies itself and outputs a TEMP FILES heading, followed by a list of temporary files on the system. This is followed by the heading FILE EQUATIONS, with a list of file equations currently being used. Here is a sample display:

```
LISTEQ5 G.02.B0 (C) HEWLETT-PACKARD CO., 1983

***TEMP FILES
INPUT.PUB.TIMONS

***FILE EQUATIONS
FILE TAPE;DEV=TAPE
FILE PRINT;DEV=LP
```

3. By default, LISTEQ5 output goes to $STDLIST, usually the terminal. The formal file designator is LIST. To redirect the output to the line printer (LP), use a file equation and run LISTEQ5 with ;PARM=1, as follows:

```
:FILE LIST;DEV=LP
:RUN LISTEQ5.PUB.SYS;PARM=1
```

Example

```
:FILE PRINT;DEV=LP
:BUILD INPUT;REC=40,3,F,ASCII;TEMP
:RUN LISTEQ5.PUB.SYS

LISTEQ5 G.02.B0 (C) HEWLETT-PACKARD CO., 1983

***TEMP FILES
INPUT.PUB.TIMONS

***FILE EQUATIONS
FILE PRINT;DEV=LP

END OF PROGRAM
:
```

LISTEQ2/LISTEQ5 Error Messages

These are the error messages you may encounter when you use LISTEQ2/LISTEQ5.

Error Message	Action
FAILURE TO GET JDT	LISTEQ5 found a job directory table of unexpected segment size. Be sure you have the current version of LISTEQ2/LISTEQ5.
FAILURE TO OPEN LIST FILE	The list file requested cannot be opened. Be sure the list file is specified correctly.
FAILURE TO WRITE ON LIST FILE	Check your file equations for device class names.

3

LISTDIR2/LISTDIR5

LISTDIR2/LISTDIR5 provides information similar to, but more detailed than, the MPE LISTACCT, LISTGROUP, and LISTF commands. Some of the information available from LISTDIR5 is accessible only to the system manager or account manager.

The program LISTDIR5.PUB.SYS is found only on MPE V/E systems; the MPE V/P version is LISTDIR2.PUB.SYS. The dialogue and operation of the two are identical. Throughout this chapter, LISTDIR5 is used for all examples and explanations.

The LISTDIR5 program allows you to do the following:

- List the attributes of accounts, users, groups, and files.
- List the security provisions for one or more files.
- List the syntax rules for all LISTDIR5 commands.

Operation

1. To run LISTDIR5, enter:

   ```
   :RUN LISTDIR5.PUB.SYS
   ```

2. LISTDIR5 identifies itself and displays the subsystem prompt (>):

   ```
   LISTDIR5 G.03.00 (C) HEWLETT-PACKARD CO., 1983

   TYPE 'HELP' FOR AID
   >
   ```

3. Enter one of the commands from the "LISTDIR5 Commands" section.

   When you run LISTDIR5 in batch mode, you must start the command in the first column of the input file. When you run LISTDIR5 interactively, you may not leave a space between the > prompt and the first character of the command.

   Standard MPE security restrictions apply to LISTDIR5. Refer to "Security Restrictions Within LISTDIR5" further on in this chapter for details.

   If you do not have sufficient capability, restricted information is replaced by double asterisks (**). This includes passwords, lockwords, disk addresses, and creator identities.

4. Press CTRL Y to halt the execution of a LISTDIR5 command. To suspend LISTDIR5 when a command is not executing, press BREAK. You are prompted for an MPE command. Type "RESUME" to continue executing LISTDIR5 after the BREAK.

By default, LISTDIR5 dialog takes place on the terminal. The formal file designator of the output file is OUT. To redirect the output, use a file equation, and use ;PARM=1 in your RUN command. For example:

```
:FILE OUT; DEV=LP
:RUN LISTDIR5.PUB.SYS;PARM=1
```

Commands

The LISTDIR5 commands are listed below and described in the following pages. The commands are:

LISTACCT	Lists the characteristics of an account.
LISTGROUP	Lists the characteristics of a group.
LISTUSER	Lists the characteristics of a user.
LISTF	Lists the attributes of a file or file set.
LISTSEC	Lists the security provisions and capabilities of a file or file set and the access rights of the user to that file or file set.
MOUNT	Generates a request at the system console to mount a private volume.
DISMOUNT	Generates a request at the system console to dismount a private volume.
EXIT	Terminates LISTDIR5.
HELP	Accesses the LISTDIR5 HELP facility.

Security Restrictions Within LISTDIR5

To prevent disclosure of group and account names and passwords, LISTDIR5 operates according to MPE security regulations. The following rules apply:

- System managers may specify any account, group, and user on the system.
- Account managers may specify any group or user name in their logon account.
- Standard users (who are not system managers or account managers) may specify only their own logon account, group, and user name.
- Passwords, lockwords, creator identities, file label addresses, and privileged file codes are displayed only when ;PASS is specified by qualified users.

- Account passwords may be listed only by system managers.
- Group and user passwords may be listed only by account or system managers.
- File lockwords and creator names may be listed by account or system managers only.
- Disk file addresses and extent maps may be displayed by the creator of the file, or an account or system manager.
- Privileged file codes may be displayed by an account manager, system manager, or the creator of the file if the creator has privileged mode capability.
- A group name containing wildcard characters may be used only by an account or system manager; refer to "Generic Names and Wildcard Characters" in this section. Standard users can use "@" for their file names.
- An account name containing wildcard characters may be specified only by a system manager.

Generic Names and Wildcard Characters

You can input file names, user names, group names, and account names as generic names. Thus you can request information on all items that meet a set of criteria, providing you are allowed access to the information. Refer to "LISTDIR5 Commands" in this section.

A generic name consists of up to eight alphanumeric and wildcard characters. Wildcard characters indicate nonspecific alphanumeric characters, so a generic name can represent several actual names. The wildcard characters are:

which represents exactly one numeric character.
? which represents exactly one alphanumeric character.
@ which represents zero or more alphanumeric characters.

A generic name must start with an alphabetic character, @, or ?. These are some examples using generic names:

`LISTF K#######.@`	Lists all files starting with "K" and followed by seven digits, in any group; for example, "K1431254", "K1418621".
`LISTF L@X`	Lists all files starting with "L" and ending with "X", with any characters in between; for example, "LAX", "LUMMOX".
`LISTF ?X@`	Lists all files whose second letter is "X"; for example, "EXTRA", "OX4".
`LISTF @`	Lists all files in the logon group.
`LISTF @.PUB.@`	Lists all files in the PUB group of all accounts. It requires system manager (SM) capability.

Private Volumes

You can use the LISTF and LISTSEC commands to list the attributes of files on private volumes if you have use volume (UV) capability. If you do not have UV capability, or if the volume class/set is not physically mounted, the relevant warnings are issued.

LISTDIR5 can list entries in the private volume directory even if there are no corresponding entries in the system directory. The MOUNT command must be used first to specify the volume set/class to be mounted, even if the MPE MOUNT command has already been entered. To do this, you must have UV capability. Subsequent LISTACCT, LISTGROUP, LISTF, and LISTSEC commands display the directory entry and file label information contained on the private volume set/class.

The account-level and group-level security displayed by these commands does not reflect access security utilized by MPE. Except for the "save" file, which is taken from a private volume directory, MPE security is always derived from the account and group entries in the system directory. The LISTUSER command always displays the entry in the system directory since there is no corresponding entry in the private volume directory.

LISTDIR5 can only mount one volume set or class at a time. Subsequent MOUNT and DISMOUNT commands cause the previously mounted volume set/class to be dismounted. The DISMOUNT command redirects LISTDIR5 to the system directory.

LISTACCT

Lists attributes of an account or set of accounts.

Syntax

LISTACCT [*acctname*] [,*listfile*] [;PASS]

Parameters

acctname	A one- to eight-character account name or generic name. The default is the logon account.
listfile	Used to specify the file or device that is to receive the output listing. It must be an existing temporary or permanent disk file, or, if preceded by *, may backreference a file equation. The default is $STDLIST.
PASS	Displays the password of the account.

Examples

```
>LISTACCT FUDD
*****************
ACCOUNT: FUDD

DISC SPACE; 4143(S)                    PASSWORD:  **
CPU TIME:  231 (SEC)                   LOC ATTR:  %0
CONNECT TIME: 232(MIN)                 SECURITY--READ:     AC
DISC LIMIT: UNLIMITED                            WRITE:    AC
CPU LIMIT:  UNLIMITED                            APPEND:   AC
CONNECT LIMIT: UNLIMITED                         LOCK:     AC
MAX PRI: BS                                      EXECUTE:  AC
USER PASSWORD REQUIRED:  OFF
GRP INX PTR:  %4346
USR INX PTR:  %4347
CAP: AM,AL,GL,DI,OP,CV,UV,CS,ND,SF,IA,BA,PH,DS,MR,PM
```

LISTGROUP

Lists attributes of a group or set of groups.

Syntax

```
LISTGROUP [groupname[.acctname] ] [,listfile][;PASS]
```

Parameters

groupname	A one- to eight-character group name or generic name. The default is the logon group.
acctname	A one- to eight-character account name or generic name. The default is the logon account.
listfile	Used to specify the file or device that is to receive the output listing. It must be an existing temporary or permanent disk file, or, if preceded by *, it may backreference a file equation. The default is $STDLIST.
PASS	Displays the password of the group.

Examples

```
>LISTGROUP @.FUDD

**********
GROUP: PUB.FUDD

DISC SPACE:  4143(S)                    PASSWORD:  **
CPU TIME: 231(SEC)                      SECURITY:--READ:      ANY
CONNECT TIME: 232(MIN)                             WRITE:     AC,GU
DISC LIMIT: UNLIMITED                              APPEND:    AL,GU
CPU LIMIT: UNLIMITED                               LOCK:      AL,GU
CONNECT LIMIT: UNLIMITED                           EXECUTE:   ANY
FILE INX PTR: %4350                                SAVE:      AL,GU
MVTABX: %0                              PRIV VOL:  NO
MOUNT REF CNT:  0
HOME VOL SET:
CAP: IA,BA,PH,DS,MR,PM
```

LISTUSER

Lists attributes of a user or set of users.

Syntax

LISTUSER [*user*[*.acctname*]] [*,listfile*] [;PASS]

Parameters

user	A one- to eight-character user name or generic name. The default is the logon user.
acctname	A one- to eight-character account name or generic name. The default is the logon account.
listfile	Used to specify the file or device that is to receive the output listing. It must be an existing temporary or permanent disk file, or, if preceded by *, it may backreference a file equation. The default is $STDLIST.
PASS	Displays the password of the user.

Examples

```
>LISTUSER ELMO;PASS

******************
USER: ELMO.FUDD

HOME GROUP: PUB                 PASSWORD: BUNNY
MAX PRI: BS,CS                  LOC ATTR: %0
PASSWORD REQUIRED: OFF
LOGON CNT:  1
CAP: AM,AL,GL,DI,OP,CV,UV,CS,ND,SF,IA,BA,PH,DS,MR,PM

LOGON GROUP: PUB                SESSION #: S768   **Appears if you are**
LOGON DEV/#: 180                                  **the current user.**
```

LISTF

Lists attributes of a file or set of files.

Syntax

LISTF *fileset* [,*listfile*] [;PASS] [;MAP]

Parameters

fileset Specifies the set of files to be listed. This positional parameter has the form:

filename[.*groupname*[.*accountname*]]

The characters "@", "#", and "?" can be used as wildcard characters in any position of the *fileset* parameter. They count toward the eight-character limit for group, account, and file names. These wildcard characters have the following meanings:

@	Specifies one or more alphanumeric characters. When used by itself, "@" denotes all members of the set.
#	Specifies one numeric character.
?	Specifies one alphanumeric character.
	These characters can be used with files as follows:
N@	Lists all files starting with the character "n".
@N	Lists all files ending with the character "n".
N@X	Lists all files starting with character "n" and ending with the character "x".
N#######	Lists all files starting with character "n" and followed by seven digits.
?N@	Lists all files whose second character is "n".
N?	Represents all two-character files starting with the character "n".
?N	Represents all two-character files ending with the character "n".

listfile The name of the output file to which the descriptions are written. It is automatically specified as a temporary ASCII file with variable-length records and these characteristics: closed in the temporary domain, CCTL (user-supplied carriage-control characters), OUT access mode, and EXC (exclusive access) option. The remainder of its characteristics are those obtained with the FILE command default specifications.

PASS Displays the lockword of the file.

MAP — Creates a list of the disk addresses of each file extent. You must be the file's creator or have account manager (AM) or system manager (SM) capability to use this parameter. The format of this map is a 32-bit number expressed in octal. The first 8 bits (0-7) are the volume table index of the logical device number upon which the extent resides. The next 24 bits (8-31) are the logical sector address.

Examples

```
>LISTF MYFILE;PASS;MAP

**********
FILE: MYFILE.PUB.FUDD

FCODE:  0                       FOPTIONS:  BINARY,FIXED,STD
BLK FACTOR: 1                   CREATOR:   ELMO
REC SIZE: 256(B)                LOCKWORD:  SECRET
BLK SIZE: 128(W)                SECURITY--READ:   ANY
EXT SIZE: 128(S)                WRITE:  ANY
# REC:  0                       APPEND: ANY
# SEC:  128                     LOCK:   ANY
# EXT:  1                       EXECUTE:     ANY
MAX REC:  1023                  **SECURITY IS ON
MAX EXT:  8                     COLD LOAD ID:  %14157
# LABELS:    0                     CREATED:  THU, 20 OCT 1983
MAX LABELS: 0                   MODIFIED: THU, 20 OCT 1983
DISC DEV#:  2                   ACCESSED: THU, 20 OCT 1983
DISC TYPE:  0                   LABEL ADR:  %1434601
DISC SUBTYPE:  9                SEC OFFSET:  %1
CLASS:  DISC                    FLAGS:  NO ACCESSORS
FCB VECTOR:  %0
EXT MAP:  %301434601
```

The abbreviations S, B, and W stand for sectors, bytes, and words, respectively. The COLD LOAD ID is a counter that keeps track of system cold loads, and helps identify the status of files when a dump is taken. LABELS and MAX LABELS refer to allocated and written user labels, not the standard file label written by MPE. SEC OFFSET indicates the number of sectors between the file label and the first data sector within the file. FLAGS indicates if and why a file has been locked down for exclusive access by a program. FCB VECTOR and INX PTR are useful to system analysts for debugging. STD, MSG, CIR, and KSAM stand for standard, message, circular, and keyed sequential access method files, respectively. All numbers are decimal unless preceded by a percent sign (%): these are octal.

LISTSEC

Lists security provisions of a file or set of files.

Syntax

LISTSEC *fileset* [,*listfile*] [;PASS]

Parameters

fileset Specifies the set of files to be listed. This positional parameter has the form:

filename[.*groupname*[.*accountname*]]

The characters "@", "#", and "?" can be used as wildcard characters in any position of the *fileset* parameter. The wildcard characters count toward the eight-character limit for group, account, and file names. These wildcard characters have the following meanings:

@	Specifies one or more alphanumeric characters. When used by itself, "@" denotes all members of the set.
#	Specifies one numeric character.
?	Specifies one alphanumeric character.
	These characters can be used with files as follows:
N@	Lists all files starting with the character "n".
@N	Lists all files ending with the character "n".
N@X	Lists all files starting with character "n" and ending with the character "x".
N#######	Lists all files starting with charcter "n" and followed by seven digits.
?N@	Lists all files whose second character is "n".
N?	Represents all two-character files starting with the character "n".
?N	Represents all two-character files ending with the character "n".

listfile Used to specify the file or device that is to receive the output listing. It must be an existing temporary or permanent disk file, or, if preceded by *, may backreference a file equation. The default is $STDLIST.

PASS Displays the creator and lockword of the file.

If an access control definition (ACD) exists for the file, the access modes displayed are those that the ACD grants you. In this case, the message ACD EXISTS is displayed.

Examples

```
>LISTSEC MYFILE
***************
FILE:   MYFILE.PUB.FUDD
SYSTEM
SECURITY--READ:      AC
  (ACCT)  WRITE:     AC
          APPEND:    AC
          LOCK:      AC
          EXECUTE:   AC
SYSTEM
SECURITY--READ:      ANY
(GROUP)   WRITE:     AL,GU
          APPEND:    AL,GU
          LOCK:      AL,GU
          EXECUTE:   ANY
          SAVE:      AL,GU

SECURITY--READ:      ANY                      FCODE:    0
  (FILE)  WRITE:     ANY                      CREATOR:  **
          APPEND:    ANY                      LOCKWORD: **
          LOCK:      ANY                    **SECURITY IS ON
          EXECUTE:   ANY

FOR ELMO.FUDD:  READ, WRITE, APPEND, LOCK, EXECUTE    **Printed if you are**
                                                      **the creator.**
```

MOUNT

Generates a request at the system console to mount a private volume.

Syntax

```
MOUNT [ { vsname } [ .groupname[ .acctname] ] ] [ ;GEN=[ genindex] ]
      { *     }
```

Parameters

vsname or *	The volume set name to be listed; no generic volume set names are permitted. "*" indicates the home volume set (default).
groupname	A one- to eight-character group name or generic name. The default is the logon group.
acctname	A one- to eight-character account name or generic name. The default is the logon account.
genindex	The generation index of the volume set or class to be mounted. The range is 0 to 65535 (default), which indicates that any generation is permitted.

DISMOUNT

Generates a request at the system console to dismount the currently mounted private volume set.

Syntax

```
DISMOUNT
```

Parameters

None.

EXIT

Terminates the LISTDIR5 subsystem.

Syntax

```
EXIT
```

Parameters

None.

HELP

Accesses the LISTDIR5 HELP facility.

Syntax

```
HELP
```

Parameters

None.

Examples

```
>HELP
CONTROL-Y MAY BE TYPED ANYTIME TO STOP THE OUTPUT FROM ANY COMMAND.

COMMAND SYNTAX IS:
  LISTACCT  [<ASET>] [,<LISTFILE>] [;PASS]
  LISTGROUP [<GSET>] [,<LISTFILE>] [;PASS]
  LISTUSER  [<USET>] [,<LISTFILE>] [;PASS]
  LISTSEC    <FSET>  [,<LISTFILE>] [;PASS]
  LISTF      <FSET>  [,<LISTFILE>] [;PASS] [;MAP]
  HELP      [<LISTFILE>]
  MOUNT     [<VDESIG>] [;GEN=[<GENINX>]]
  DISMOUNT
  EXIT

  <ASET>     IS AN ACCOUNT NAME.
  <GSET>     IS A GROUP NAME, OPTIONALLY QUALIFIED BY AN
             ACCOUNT NAME.
             EXAMPLES:  LISTGROUP MYGROUP.MYACCT
                        LISTGROUP @.MYACCT
                        LISTGROUP @.@
  <FSET>     IS A FILE NAME, OPTIONALLY QUALIFIED BY A GROUP AND ACCOUNT NAME.

CONTINUE? (Y/N)N
>
```

LISTDIR2/LISTDIR5 Error Messages

These are the error messages you may receive from LISTDIR5.

Error Message	Action
INVALID COMMAND	Re-enter the command with the correct spelling and syntax.
NO SUCH FILE	An invalid file name was entered in the command. Check your spelling or do a LISTF to see if this file is in your account.
NO SUCH GROUP	An invalid group name was specified in the command.
NO SUCH ACCOUNT	An invalid account name was specified in the command. Check the spelling and re-enter.
NO SUCH USER	An invalid user name was specified in the command.
INVALID PARAMETER	An invalid parameter was specified in the command.
DUPLICATE PARAMETER	The same parameter was specified twice in this command.
INVALID DESIGNATOR	An invalid designator was specified in the command.
ACCOUNT IS NOT LOG ON	You must have system manager capability to gain information about an account other than your logon account.
GROUP IS NOT LOG ON	You must have account manager capability to list information about groups other than your own.
USER IS NOT LOG ON	You must have account manager capabilities to access information about other users.
DISC I/O ERROR	An I/O error occurred while trying to access the directory.
MISSING PARAMETER	Check the parameter syntax and re-enter.
INVALID FILESET FOR USER	An invalid file set for user files was specified in a command.
LOCKWORD IS NOT ALLOWED	A lockword specification is not allowed in a command.
CANNOT CLOSE LIST FILE--FSERR=*nnn*	An error was encountered in closing the list file. Check the FSERR number.
CANNOT OPEN LIST FILE--FSERR=*nnn*	An error was encountered in opening the list file. Check the FSERR number.
WRITE ERROR ON LIST FILE--FSERR=*nnn*	A write error was encountered on the list file. Check the FSERR number.
INVALID LIST FILE NAME	Check the file name to be sure that it conforms to MPE naming conventions. If you are using a file equation, be sure that the device class in the equation can be used for listing, and that the list file name begins with a "*".

Error Message	Action
FILE NAME BEGINS WITH NUMERIC CHARACTER OR #	The first character of the file name must begin with a letter.
MISSING FILE NAME	The file name specification is missing from this command. Check the syntax and re-enter.
FILE NAME EXCEEDS 8 CHARACTERS	Check the file name and retry.
MISSING DELIMITER AFTER FILE NAME	Check the command syntax and re-enter.
GENERIC FILE NAME IS NOT ALLOWED	You cannot specify this file with a wildcard character.
GROUP NAME BEGINS WITH NUMERIC CHARACTER OR #	The first character of this group name must be a letter.
MISSING GROUP NAME	Check the command syntax and re-enter.
GROUP NAME EXCEEDS 8 CHARACTERS	Change the group name and re-enter.
MISSING DELIMITER AFTER GROUP NAME	Check the command syntax and re-enter.
GENERIC GROUP NAME IS NOT ALLOWED	A group name containing wildcard characters may be specified only by a system manager or account manager.
ACCOUNT NAME BEGINS WITH NUMERIC CHARACTER OR #	The first character of the account name must be a letter.
MISSING ACCOUNT NAME	Check the command syntax and re-enter.
ACCOUNT NAME EXCEEDS 8 CHARACTERS	Check your spelling and re-enter the name.
MISSING DELIMITER AFTER ACCOUNT NAME	Check the command syntax and re-enter.
GENERIC ACCOUNT NAME IS NOT ALLOWED	An account name containing wildcard characters may be specified only by a system manager. No wildcard characters are permitted in the *acctname* parameter of the MOUNT command.
USER NAME BEGINS WITH NUMERIC CHARACTER OR #	The first character of the user name must be a letter. Correct the name and re-enter.
MISSING USER NAME	Check the command syntax and re-enter.
USER NAME EXCEEDS 8 CHARACTERS	Check your spelling and re-enter.

Error Message	Action
MISSING DELIMITER AFTER USER NAME	Check the syntax for the LISTUSER command.
GENERIC USER NAME IS NOT ALLOWED	You may specify only your logon user name unless you are a system manager or account manager.
VOLUME NAME BEGINS WITH NUMERIC CHARACTER OR #	The volume name must begin with a letter.
MISSING VOLUME NAME	Either a volume set name or a backreference to the home volume set is required in the MOUNT command.
VOLUME NAME EXCEEDS 8 CHARACTERS	Check spelling and re-enter the name.
MISSING DELIMITER AFTER VOLUME NAME	Check the command syntax of the MOUNT command.
GENERIC VOLUME NAME IS NOT ALLOWED	No wildcard characters are allowed in the *vsname* parameter of the MOUNT command.
USER DOES NOT HAVE UV CAPABILITY	You must have use volumes (UV) capability to access a private volume.
PRIVATE VOLUME ERROR=*nnn*	An error was encountered in accessing a private volume.
COMMAND IS NOT IMPLEMENTED	You have specified a command that is not functional in the current version of MPE.
INVALID GENERATION INDEX	The value you specified for *genindex* in the MOUNT command must be between 0 and 65535.
MOUNT REQUIRED BEFORE LISTING FILES ON PRIVATE VOLUME	You cannot use LISTUSER, LISTSEC, or LISTF until after you have specified the volume set or class to be mounted with MOUNT.
MPE COMMANDS MAY NOT BE EXECUTED FROM LISTDIR5	You must exit the system before attempting to execute any MPE commands.
ERROR FOUND ON INPUT. LISTDIR5 TERMINATED	An error was encountered in reading the input string.
EOF DETECTED. LISTDIR5 TERMINATED	An end-of-file condition was encountered in reading the input string.
EOF DETECTED ON OUTPUT FILE	An end-of-file condition was encountered in writing to the list file.

4

SPOOK/SPOOK4/SPOOK5

The SPOOK, SPOOK4, and SPOOK5 utility programs allow you to list, manipulate, and transfer spooled device files (spool files) created and maintained by MPE. Any user can access this utility, but some functions are limited to users with privileged mode (PM), system manager (SM), or account manager (AM) capabilities.

SPOOK is the version of the utility supplied with MPE IV and MPE V/P (E/F.00.00). Both SPOOK4 and SPOOK5 are supplied with MPE V/E. The command sets for all three versions are identical. If the system manager has established a system-wide user-defined command (UDC) for the appropriate version of SPOOK, as described below, most users will be unaware of any changes. For the remainder of this chapter, the term "SPOOK" is used as a generic reference to all three versions, unless indicated otherwise.

Operation

The version of SPOOK you use depends on the operating system running on your computer. Refer to "Multi-Version and Upgraded Sites" in this section for a discussion of special cases when SPOOK4 should be used instead of SPOOK.

MPE V/P and MPE IV Systems

If you are on MPE V/P or MPE IV, or if you have several systems running any combination of MPE V/P (E/F.00.00) and MPE IV, you use the program SPOOK.PUB.SYS. The SPOOK tapes from both versions of MPE are completely compatible. To simplify the use of SPOOK, the system manager can establish the following system-wide UDC:

```
SPOOK
COMMENT  *** This UDC is for an MPE V/P or MPE IV system ***
RUN SPOOK.PUB.SYS
***
```

MPE V/E Systems

If you have one or more systems running only MPE V/E, you use the program SPOOK5.PUB.SYS. Any MPE V/E system can read SPOOK5 tapes. The system manager can establish the following system-wide UDC:

```
SPOOK
COMMENT  *** This UDC is for an MPE V/E system ***
RUN SPOOK5.PUB.SYS
***
```

Multi-Version and Upgraded Sites

If your site has systems that run MPE V/E as well as MPE V/P (E/F.00.00) and/or MPE IV, you need to use SPOOK5 on your MPE V/E system and SPOOK4 on your other system.

The format of the SPOOK tape includes system table entry images of the device class table and the output device directory, both of which changed in MPE V/E. For this reason, SPOOK tapes from MPE IV and MPE V/P have a different format from the SPOOK tapes of MPE V/E. The spool files themselves remain the same in all versions.

SPOOK5, which runs only on MPE V/E systems, is able to read MPE IV, MPE V/P (E/F.00.00), and MPE V/E SPOOK tapes. Therefore, all present or past SPOOK tapes from any of your systems can be read by SPOOK5. For convenience, the system manager can establish a UDC for SPOOK5 as described above.

If you wish to transfer spool files created on an MPE V/E system to an MPE V/P (E/F.00.00) or MPE IV system, the program `SPOOK4.PUB.SYS` must reside on the MPE V/P or MPE IV system. For convenience, SPOOK4 is shipped with every MPE V/E system, even though it is not run on MPE V/E. To transfer SPOOK4 to your non-MPE V/E system, store `SPOOK4.PUB.SYS` from your MPE V/E system onto tape, and restore it onto the MPE V/P (E/F.00.00) or MPE IV system. The system manager can establish a `SPOOK` UDC to `RUN SPOOK4.PUB.SYS`, allowing users to use the same command to run SPOOK on all systems.

Commands

These are the commands you can use in the `SPOOK` subsystem.

`SHOW`	Lists characteristics of input/output spool files.
`TEXT`	Accesses (opens) an output spool file to permit listing its contents (`LIST` command) or locating records within it (`FIND` command).
`LIST`	Lists the contents of all or part of an output spool file made accessible by the `TEXT` command.
`FIND`	Locates a character string in an output spool file accessed by the `TEXT` command, and lists the first record containing the string.
`PURGE`	Deletes an output spool file.
`APPEND`	Appends part or all of an output spool file to a new output spool file until the "append" ends.
`COPY`	Copies part or all of an output spool file to another output spool file.
`MODE`	Modifies the maximum width and/or requests a display of control information for lines listed by `LIST` or `FIND` commands.

ALTER	Changes the output priority, number of copies requested, and/or destination device for an output spool file.
EXIT or QUIT	Terminates the operation of SPOOK.
HELP or XPLAIN	Lists all SPOOK commands and describes their syntax.
DEBUG	Invokes the MPE debug facility. You must have privileged mode (PM) capability.
OUTPUT	Stores output spool files on tape or serial disk. You must have system manager (SM) or system supervisors (OP) capability.
INPUT	Restores output spool files from serial storage to the system. You must have system manager (SM) or system supervisors (OP) capability.
RUN	Executes a prepared program.
KILL	Deletes a child process of SPOOK.

Any MPE command that can be accessed programmatically by the COMMAND intrinsic can be accessed by SPOOK. For a complete list of these MPE commands, refer to the *MPE V/E Commands Reference Manual* (32033-90006).

Running SPOOK

To run SPOOK5, type the following;

```
:RUN SPOOK5.PUB.SYS
```

The program identifies itself and displays a > prompt:

```
SPOOK G.03.00  (C)  HEWLETT-PACKARD CO., 1983.
```

Entering SPOOK Commands

The following pages list syntax, parameters, operation, and examples of the SPOOK commands. Each command can contain up to 80 characters. Blanks may appear anywhere between syntactic elements such as names, separators, or keywords, but not within a command name. A qualified file name or user name, such as *username.acctname,* cannot contain embedded blanks.

On most terminals, you press CTRL X to delete the current line, and 'CTRL Y to terminate the current operation.

Spool files are usually referenced by device file ID or by owner. The device file ID is a unique identifier in the form #I*nnn* for input files, and #O*nnn* for output files. (The #O is optional for output spool files.) The owner is denoted by the *username.acctname* of the creator, as reflected in the file label.

SPOOK restricts access to spool files according to the user's logon capability, as follows:

- System managers or system supervisors can access any spool file.
- Account managers can access any spool file in the logon account.
- Standard users can access any spool file that they created.

Syntax for a Range List

When a SPOOK function requires you to enter a range, you may specify `ALL`, or use the following syntax to specify a subset of the file:

[,*count*]

[{ *recnumber1* / * / FIRST / LAST } [±*offset*]] [/ { *recnumber2* / * / FIRST / LAST } [±*offset*]]

where:

recnumber	Is an absolute record number, or a line of text in the file.
*	Specifies the current line.
offset	Specifies a record number relative to *recnumber*.
count	Specifies a number of lines in the file.

Examples of Range Lists

```
>L 58/58+19
>L 22/LAST
>L FIRST/LAST
>L ALL
>L FIRST+22/LAST
>L 29/82
```

ALTER

Changes output priority, number of copies requested, or destination device for the output spool file.

Syntax

```
>ALTER { username[.acctname] dfid[,dfid][, ... ]}

{ ;PRI=outpri          } [ ,PRI=outpri          ]
{ ;COPIES=copies       } [ ,COPIES=copies       ] [, ... ]
{ ;DEV= { ldev       } } [ ,DEV= { ldev       } ]
{       { devclass   } }   [       { devclass   } ]
```

Parameters

username	Name of creator of file to be altered, or "@" to alter all files of all users.
acctname	Account name of creator of file to be altered, or "@" to alter files of all accounts.
dfid	Identifier of output spool file to be altered, in the form `#0`*nnn* (`#0` is optional). To reference the current spool file accessed by the `TEXT` command, enter an asterisk (*).
outpri	The intended priority of the spool file being altered, from 1 (lowest priority) to 13 (highest priority). When `PRI=1`, output is always deferred. Use `PRI=2` or greater if you want the spool file to be scheduled for output.
copies	An integer from 1 to 127, indicating the number of copies to be printed.
ldev	The logical device number of the device on which the spool file is printed.
devclass	The device class name of the device on which the spool file is to be printed.

Note

For information about device ACDs, see chapter 4, "Managing Access to MPE V/E Files", in the *MPE V Security Management Guide.*

Operation

The ALTER command can be abbreviated as A.

The parameters PRI, COPIES, and DEV can be abbreviated as P, C, and D, respectively. If you omit any PRI, COPIES, or DEV parameters, the corresponding attribute of the spool file remains unchanged.

ALTER operates only on output spool files in the READY, OPEN, or LOCKED states, or a file currently accessed by the TEXT command.

Examples

To change the currently scheduled number of output copies to 3 for spool file #025, enter:

```
>ALTER #025; COPIES=3
```

To change the output priority of a spool file currently accessed by the TEXT command to 7, and the destination device to the device with the class name LP, enter:

```
>A *; P=7, D=LP
```

To change the output priority of all spool files to 8, enter:

```
>A @.@;P=8
```

To change the priority of all spool files from the PAYROLL account to 6, enter:

```
>A @.PAYROLL; P=6
```

APPEND

Appends all or part of a spool file to another file.

Syntax

```
>APPEND  [range[,filename]]
```

or

```
>APPEND [ { username[.acctname] } [;range[,filename] ] ] [END]
        { dfid[,dfid][,...]   }
```

Parameters

username	Name of the user who created the file(s) to append.
acctname	Account name of the user who created the file(s) to append.
dfid	Identifier of the output spool file to append; in the form `#O`*nnn*. (`#O` is optional).
range	Specific line numbers of the file to append. Refer to "Syntax for a Range List" at the beginning of this section.
filename	The spool file to which the file is to be appended; it can be `$STDLIST`, or it can refer to a file equation if in the form **filereference*. If *filename* is omitted, and no previous *filename* is currently in effect, `APPEND` opens and names a new spool file.
`END`	Closes the new file.

Operation

The `APPEND` command joins part or all of a spool file to another file. `APPEND` opens a new spool file if *filename* is omitted the first time `APPEND` is used. The same file remains open and subject to appending until an `APPEND END` command, `COPY` command, or termination of SPOOK. `APPEND` can be abbreviated `APPEN`, `APPE`, `APP`, or `AP`. You can use the first format, `APPEND [`*range*`[,`*filename*`] ]`, only after the file has been texted.

Caution

When appending a spool file or range of spool files to a permanent file, use only spool files that do not include environmental information. The SPOOK utility does not append environmental information correctly to permanent files.

Examples

To append two spool files:

```
>T #021          **Text in output file number 21.**
>APPEND ALL      **The new spool file is still open.**
>T #022          **Text in output file number 22.**
>AP ALL          **The next spool file is appended.**
>APPEND END      **The new spool file is closed.**
>PURGE 21,22     **Purge originals.**
```

To append all files created by BRUCE.MPEUTIL, and have them put on tape:

```
>FILE TAPE; DEV=TAPE
>APPEND BRUCE.MPEUTIL; ALL, *TAPE
>APPEND END
```

To append #O22 to #O21:

```
>APPEND #021, #022; ALL
>APPEND END
```

Note Most MPE commands are available in SPOOK.

COPY

Copies all or part of a spool file to another file.

Syntax

```
>COPY  [ range[ filename ]   ]
```

or

```
>COPY  [ { username[ .acctname ] }  ]  [ ; range[ ,filename ]  ]
       [ { dfid[ ,dfid ][ , ... ] }  ]
```

Parameters

username	Name of the user who created the file(s) to copy, or "@" to copy files of all users.
acctname	Account name of the user who created the file(s) to copy, or "@" to copy files of all accounts.
dfid	Identifier of output spool file to copy, in the form `#O`*nnn*. The `#O` is optional.
range	Specific line numbers of the file to copy. Refer to "Syntax for a Range List" at the beginning of this section.
filename	The spool file to which the file is to be copied; it can be `$STDLIST`, or it can refer to a file equation if in the form **filereference*. If *filename* is omitted, and no previous *filename* is currently in effect, `COPY` opens and names a new spool file.

Operation

The `COPY` command copies part or all of a spool file to another file. It closes the new file when finished. `COPY` can be abbreviated as `COP`, `CO`, or `C`.

If no *username* or *dfid* is specified, this function copies the currently texted file to a new file.

The first format, `COPY [`*range*`[`*filename*`] ]`, can be used only after the file has been texted.

Caution

When copying a spool file or range of spool files to a permanent file, you can only use spool files that do not include environmental information. The SPOOK utility does not copy environmental information correctly to permanent files.

Examples

To copy spool file #O123 to a permanent file called REPORT, do the following:

```
>FILE REPORT;DEV=DISC;REC=-80,,F,ASCII
>T #0123
>COPY ALL,*REPORT
```

To split one file into two and purge the original, do the following:

```
>SHOW 293            **Determine the file's length.**
#FILE   #JOB      FNAME    STATE  DEV/CL   PR COP RFN OWNER
#0293   #S1303    LOADMAP  LOCKED LP       8  1       USER.ACCT
#FILE   LDEV      LABEL      SECTORS       LINES      TIME
#0293   %5       %531461       24          77         9:50  7/7/82

>TEXT  #0293          **Text in the source spool file.**
>COPY FIRST/36
>COPY 37/LAST

>SHOW             **Verify the creation of the split files.**
#FILE    #JOB      FNAME      STATE    OWNER
#0293    #S1303    LOADMAP    LOCKED   USER.ACCT
#0295    #S1303    LOADMAP    READY    USER.ACCT
#0297    #S1303    LOADMAP    READY    USER.ACCT

>PURGE *          **Purge texted file.**
#FILE    #JOB      DEV/CL        SECTORS      OWNER
#0293    #S1303    LP                 24      USER.ACCT
```

To copy #O21 and #O22 to new spool files, do the following:

```
>COPY #021, #022;  ALL
```

To add environment file information to spool file #O531, use COPY to create a new spool file:

```
>FILE Q; DEV=HP2680; ENV=envfilename
>COPY #0531;ALL,*Q
```

You may replace HP2680 with any device class name that specifies the device using the environment file, depending upon the configuration of your system.

When you use the COPY command for copying multiple files onto labeled tapes, SPOOK copies all the files with one tape request. When you use unlabeled tapes, SPOOK generates multiple tape requests, and rewinds the tape after each file is copied.

DEBUG

Invokes the MPE Debug facility.

Syntax

```
>DEBUG
```

Operation

The DEBUG command is used primarily by system programmers. DEBUG allows programmers to set breakpoints within programs, and display and modify data stacks and registers. DEBUG can set breakpoints within the SPOOK program and manipulate SPOOK's data stack. Refer to the *MPE Debug/Stack Dump Reference Manual* (30000-90012) for details.

DEBUG is available to you only if you have privileged mode (PM) capability. When you access it from SPOOK, DEBUG runs in privileged mode.

Caution

The normal safeguards of MPE are bypassed in privileged mode. When you attempt to modify privileged data on disk, it is possible to destroy file integrity, or the MPE operating system itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged data, but this service is not included in the standard service contract. Hewlett-Packard will not modify the MPE operating system to accommodate problems arising from your use of DEBUG.

EXIT

Terminates the operation of SPOOK.

Syntax

```
>EXIT
```

Operation

The `EXIT` command closes any open text file and terminates SPOOK. If this execution of SPOOK is a child process, SPOOK is suspended and you are returned to the parent process.

FIND

Locates a character string in a file.

Syntax

```
>FIND [@] ["string"][,range]
```

Parameters

@	Specifies that entire lines are to be searched, rather than just leading characters.
string	The search key to be located by FIND. May be any number of alphanumeric characters, enclosed in quotation marks. If you omit it, FIND locates the first record (line) of the file.
range	The area of the file in which to conduct the search. Refer to "Syntax for a Range List" at the beginning of this section for more information.

Operation

The FIND command scans the file previously accessed by TEXT and locates the first record containing a string matching the *string* parameter. SPOOK can scan selected parts of the file. Leading blanks and control characters in the *string* parameter and file records are ignored. When the string is found, FIND displays the record containing the string and the record number. If the string is not found, FIND sets its current record pointer to one record beyond the last record scanned.

For each record located, FIND displays the following:

- Line number. The record number where the string was found.
- I/O Control. I/O controls applying to the file, displayed only if requested by the MODE command. *Wnnn* indicates *write with control nnn;* *Cnnn* indicates *control using nnn;* FOPEN indicates an open file; and FCLOSE indicates a closed file.
- Text. Any leading control character is removed; nonprinting characters are each replaced by a period. If the display length has been reduced with the MODE command, the record text might appear truncated.

```
Line Number  I/O Control      Text (line contents)
     |          |                 |
     7       W%320  And there they met the Beast, a ponderous ...
```

Examples

To search spool file #O250 for the record with leading characters NO. ERRORS:

```
>TEXT #0250
>FIND "NO. ERRORS", ALL
```

Now, within records 21 to 30 of the same file, find the first record with ":" appearing anywhere in the line:

```
>F @ ":", 21/30
```

Now, from the current record to the end, scan leading characters for :EOD:

```
>F ":EOD", */last
```

HELP

Lists and describes SPOOK and MPE commands.

Syntax

```
>HELP [ { MPE                    } ]
        { HELP                   }
        { tablecontents          }
        { command, [keyword]     }
        { ALL                    }
```

Parameters

MPE	SPOOK enters the MPE HELP subsystem.
HELP	Displays information about the MPE HELP facility.
tablecontents	To request general MPE information, *tablecontents* can be SESSIONS, JOBS, PROGRAMS, FILES, MANAGE, or UTILITY.
command	Can be any MPE command.
keyword	Can be PARMS, OPERATION, or EXAMPLE for information about the command parameter, operation, or an example, or ALL for all three.
ALL	Displays the table of contents of the MPE HELP facility.

Operation

The HELP command cannot be abbreviated.

If no parameters are specified, SPOOK describes its own commands.

The MPE HELP facility uses the same > prompt as the SPOOK subsystem, and you may be unsure about which one you are in. To check, enter Z (or any character which is recognized by neither). If you are in SPOOK, the INVALID COMMAND message appears. If you are in the MPE HELP facility, you see the message:

```
Can't find anything under this command or in the table of contents.
```

To exit from the MPE HELP facility and return to SPOOK, type E, EXIT, or END.

INPUT

Restores output spool file from serial storage to system.

Syntax

```
>INPUT [ username[ .acctname] ]
       [ dfid[ ,dfid][ , ... ] ]  ;*tapefile
```

Parameters

username	User name of creator of file(s) to restore, or "@" to restore the files of all users.
acctname	Account name of the creator of the file(s) to restore, or "@" to restore the files of all accounts.
dfid	Device file identification of the output file to restore, in the form `#O`*nnn*. `#O` is optional. If *username.acctname* and *dfid* are both omitted, all files belonging to the current user are restored.
tapefile	The tape file from which the input comes. A file equation must exist, and *tapefile* must be backreferenced with a "*" in the form **tapefile*.

Operation

The `INPUT` command reads files back onto the system disk that were previously stored on tape or serial disk by the SPOOK `OUTPUT` command. You must have system manager (SM) or system supervisor (OP) capability.

`INPUT` handles single and multi-reel tape files, and serial disk files. After a file is input, it is assigned a new device file ID, and the job or session number of its creator is flagged with an apostrophe as follows: `#S'`*nnn* or `#J'`*nnn*

The apostrophe indicates the file was copied into the system from an external source. The device class name and logical device number of the file change if the file is restored to a different type of device from the one on which it originated.

For each spool file restored, SPOOK displays its old and new device file IDs, the new job or session number, the new logical device/class name, and the creator, in the following form:

```
#ENTITY ===>  #FILE     #JOB     DEV/CL     OWNER
#099     ===>  #0105     #S'44    LP          BRUCE.MPEUTIL
#0100    ===>  #0106     #S'44    LP          BRUCE.MPEUTIL
```

The output priority of the new device file is automatically set to 1. You must alter the file output priority.

Examples

To restore all files from tape file T:

```
>FILE T;DEV=TAPE
>INPUT @.@;*T
```

To restore files #O3, #O4, and #O7 from the same tape:

```
>I 3,4,7;*T
```

To restore all files from the same tape belonging to BILL in the JONES account:

```
>IN BILL.JONES;*T
```

KILL

Deletes a child process.

Syntax

```
>KILL
```

Operation

The KILL command deletes the child process created by the RUN command, and all its descendants.

LIST

Lists lines of a file.

Syntax

```
>LIST [range ]
```

Parameters

range The range of lines in the file to list. Refer to the "Syntax for a Range List" for more information.

Operation

For each record, the LIST command displays:

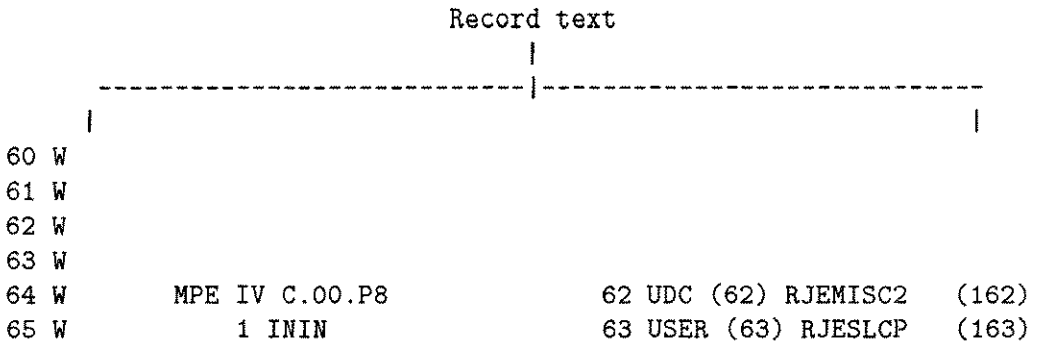
- Line number. The record number where the string was found.
- I/O Control. I/O controls applying to the file, displayed only if requested by the MODE command. W*nnn* indicates *write with control nnn;* C*nnn* indicates *control using nnn;* FOPEN indicates an open file; and FCLOSE indicates a closed file.
- Text. Any leading control character is removed; nonprinting characters are each replaced by a period. If the display length has been reduced with the MODE command, the record text might appear truncated.

This command must be preceded by a TEXT command. If not, the following error message appears:

```
>LIST
*ERROR = 46 BYTE = 4*  NO TEXT FILE.
```

If your spool file contains HP 2680 Page Printer environment file information, you must use the MODE command to enable the display of nonprinting control characters. Here is an example of environment data appended to a spool file:

```
Line      I/O Control                          Control Information
number        |                                        |
|   --------------------   ------------------------------------------------
   |                    |  |                                                |
0 FOPEN
1 FDEVICECONTROL FUNC=138 P1=% 000000 P2=% 000000 LEN=132 Download/Delete VFC
2 FDEVICECONTROL FUNC=138 P1=% 000000 P2=% 100000 LEN= 20 Download/Delete VFC
3 FDEVICECONTROL FUNC=135 P1=% 000000 P2=% 000000 LEN=132 Download/Delete Forms
4 FDEVICECONTROL FUNC=135 P1=% 000000 P2=% 100000 LEN=132 Download/Delete Forms
6 FDEVICECONTROL FUNC=135 P1=% 000000 P2=% 100000 LEN=132 Download/Delete Forms
```



When the listing is complete, the current record pointer is set to point at the record after the last listed record.

Examples

To list all records in output file #O16:

```
>TEXT #O16
>LIST ALL
```

If the same file contains environment file information:

```
>TEXT #O16
>MODE CONTROLS=ON
>LIST ALL
```

To display the current record of this same file:

```
>L
```

To list all records from two records past the current record, up to the fourth-before-last record:

```
>L *+2/LAST-4
```

To list a total of three lines, starting with record #5:

```
>L 5, 3
```

This lists lines five, six, and seven.

MODE

Adjusts line width and enables display of control data from FIND or LIST commands.

Syntax

```
>MODE [WIDTH={±nnn | OFF}] [,] [CONTROLS={ON | OFF}]
```

Parameters

±*nnn*	A signed integer indicating the width of output from FIND or LIST commands. A positive sign indicates width in words; negative sign indicates bytes.
OFF	WIDTH=OFF lets SPOOK use the actual source record width. CONTROLS=OFF disables display of I/O control characters.
ON	CONTROLS=ON enables display of I/O control characters resulting from LIST and FIND commands.

Operation

The MODE comand lets you alter the values of the WIDTH and CONTROLS parameters. The initial value of WIDTH and CONTROLS is OFF. Both settings remain in effect until changed with the MODE command.

If the spool file under observation contains HP 2680 Page Printer environment information, enter MODE CONTROLS=ON before using the LIST command.

Examples

To set the maximum line width to 60 characters (bytes), and enable display of control characters:

```
>MODE WIDTH=-60 CONTROLS=ON
```

To set the display line width to the actual record size:

```
>M W=OFF
```

Commands from the first example remain set until disabled by:

```
>M C=OFF
```

OUTPUT

Stores output spool files on tape or serial disk.

Syntax

```
>OUTPUT [ username[ .acctname ] ]
        [ dfid[ ,dfid ][ , ... ] ]  ;*tapefile [ ;PURGE ]
```

Parameters

username	User name under which the spool files to be stored were created, or "@" to store files of all users.
acctname	Account name under which the spool files to be stored were created, or "@" to store the files of all accounts.
dfid	Identifies the output spool file to store, in the form #O*nnn*. (#O is optional.)
tapefile	The file to which to store the spool files. Use "*" to backreference an existing MPE file equation.
PURGE	Indicates that files should be purged from the system after being stored.

Operation

The OUTPUT command lets the system manager or system supervisor copy output spool files onto tape or serial disk. These can be restored to the system using the INPUT command. Only READY and LOCKED spool files can be output. If you are outputting files to multiple tape reels, the system prompts you with this message:

```
CHANGE VOLUMES ON LDEV nn? YES (Y) or NO (N)
```

To change reels or disks, you enter Y and mount the new medium. To abort the operation, you enter N. If you mount the wrong medium, SPOOK displays this message:

```
INCORRECT VOLUME - TRY AGAIN? YES (Y) or NO (N)
```

As each file is copied to external storage, its output priority is set to 1 (deferring output). Before you use the OUTPUT command on that file again, use the ALTER command to raise its output priority.

For each spool file stored, SPOOK lists the current device file ID, the session or job number of the file's creator, its logical device number or class, the number of sectors in the file, and the creator's name and account, in the form:

```
#FILE       #JOB       DEV/CL      SECTORS       OWNER
#099        #S44       LP                8       BRUCE.MPEUTIL
#0100       #S44       LP               12       BRUCE.MPEUTIL
```

Before using the OUTPUT command, set up an MPE file equation with these parameters:

:FILE *formaldesignator*[=*filereference*];DEV=*device*

The ;DEV= parameter must indicate the device class name or logical device number of a magnetic tape unit or a serial disk. The OUTPUT command supplies all other parameters, such as ;REC= and ;ACC=; if you attempt to supply any of these, SPOOK rejects the OUTPUT command.

Examples

To store all output spool files created by the logon user to the tape file T:

```
>FILE T; DEV=TAPE
>OUTPUT @.@; *T
```

PURGE

Deletes output spool files from the system.

Syntax

```
>PURGE { dfid } [ dfid, ] [ , ... ]
       { *    }
```

Parameters

dfid	Identifies an output spool file in the form `#O`*nnn*. `#O` is optional.
*	Purges the file currently accessed by `TEXT`.

Operation

The `PURGE` command can be abbreviated as `P`. It deletes one or more spool files that are in the `READY` or `LOCKED` state, or the file currently accessed by `TEXT`. For each spool file purged, SPOOK lists its device file ID, its creator's session or job number, its logical device/class, the number of sectors in the file, and the creator's name and account, in the form:

```
#FILE      #JOB      DEV/CL     SECTORS      OWNER
#O367      #S156     LP               8      BRUCE.MPEUTIL
#O368      #S156     LP              12      BRUCE.MPEUTIL
#O412      #S163     EPOC            32      GUY.MPEM
```

Examples

To delete spool file #O24:

```
>PURGE #O24
```

To delete spool files #O19, #O20, and #O21:

```
>P 19,20,21
```

Note

You must be the creator of the files in question in order to purge them as in the second example above.

QUIT

Terminates the execution of SPOOK.

Syntax

```
>QUIT
```

Operation

The `QUIT` command can be abbreviated as `Q`. It closes any open text file and terminates SPOOK.

RUN

Executes a program file.

Syntax

```
>RUN [ progfile ]
     [ *        ]
```

Parameters

progfile	Designator of the prepared program file; it can be a fully qualified file name.
*	Specifies the program file previously run but now suspended.

Operation

When you run a program using the RUN command, a nonzero parameter is passed to the program to be run. You can run SPOOK, or any other program that does not accept parameters. However, do not run a program that accepts parameters, since the parameter passed will produce unpredictable results.

Also, do not use the ;PARM= parameter with the RUN command. The ;PARM= parameter is part of the MPE RUN command, not the SPOOK RUN command.

SPOOK can be executed from within itself. The prompt at each sublevel of SPOOK is numbered; for example:

```
>(1)
>(2)
```

The EXIT command suspends SPOOK, keeping its file pointer intact. Therefore, you can alternate between two active files.

QUIT causes SPOOK to terminate and return to the parent process, while KILL causes the parent process to terminate its child process.

Examples

To run a program file:

```
>RUN EDIT.TIM.DOWN
```

To run another session of SPOOK in order to use TEXT with two spool files at the same time:

```
:RUN SPOOK5.PUB.SYS

SPOOK5 G.03.00  (C) HEWLETT-PACKARD CO., 1983

>T #0111                    **Text in the first file.**
   .
   .                                **Use it and leave it texted in.**
   .
>RUN SPOOK5.PUB.SYS         **Run the nested program.**

SPOOK5 G.03.00  (C) HEWLETT-PACKARD CO., 1983
>(1)T #0222                 **Text in the second file.**
   .
   .                                **Use it.**
   .
>(1)E                       **Exit the nested program.**

SPOOK5 G.03.00  (C) HEWLETT-PACKARD CO., 1983
>E                          **Exit SPOOK**

END OF PROGRAM
:
```

SHOW

Lists characteristics of input or output spool files.

Syntax

```
>SHOW [ [ username[ .acctname] [;[@] [I] [O] ] ]
      [ dfid[ ,dfid] [ , ... ] ]                    [*]
```

Parameters

username	The user name of the creator of the file to be shown, or "@" for the files of all users.
acctname	The account name of the creator of the file to be shown, or "@" for the files of all accounts.
@	Requests the output in long format. The default format is short.
I	Requests a display of input spool files only.
O	Requests a display of output spool files only.
dfid	Identifies the spool file to be displayed, in the form #O*nnn* for output files, and #I*nnn* for input files. When *dfid* is specified, the output is always in long format.
*	Enter an asterisk (*) to refer to the current text file.

Operation

The SHOW command can be abbreviated as S.

The short format includes the device file ID, the session or job number of the creator, the file name, and the file state:

```
#FILE     #JOB     FNAME     STATE     OWNER
#0367     #S156    LISTER    READY     BRUCE.MPEUTIL
#0368     #S156    SLP       ACTIVE    BRUCE.MPEUTIL
#0369     #S156    BRUFILE   READY     BRUCE.MPEUTIL
```

The file state can be:

ACTIVE	The file is being transmitted to or from a storage input or output device.
READY	An input file is ready for use by a program, or an output file is stored on disk and ready for output.
OPENED	The input or output spool file is being accessed by a program.
LOCKED	The output spool file is being held by the system for exclusive access.

The long format appears as follows:

```
#FILE   #JOB    FNAME    STATE  DEV/CL   PR COP RFN OWNER
#02933  #J90    $STDLIST OPEN   EPOC      8   1   R MAILMAN.HPOFFICE
#03103  #S816   SLP      OPEN   PP        8   1     MGR.SOPRM
#02386  #J71    $STDLIST READY  LP        1   1     MANAGER.DOC
#FILE   LDEV    LABEL      SECTORS       LINES      TIME
#02933  %3     %5042507     2048           0
#03103  %3     %5064337     8192          68096
#02386  %3     %5666277       36           47          9:34  1/30/85
```

The long format includes everything in the short format, plus:

- The logical device number or class name of the originating device (for input files) or the destination device (for output files).
- The number of copies requested (for output files).
- RFN, where

  R indicates a restartable spooled job file,

  F indicates that a forms alignment message applies to the spool file, and

  N indicates that insufficient disk space was available when the spool file was created.
- The logical device number of the device where the file label is stored.
- The sector address of the spool file label.
- The total number of disk sectors used by the file.
- The total number of records (lines) in the spool file.
- The time the spool file entered the READY state.

Examples

To display, in long format, the characteristics of all spool files created by SMITH.ALPHA:

```
>SHOW SMITH.ALPHA; @
```

To display, in short format, the characteristics of all spool files created by the current user only:

```
>S
```

To show the characteristics of all output spool files:

```
>S @.@; O
```

To show the characteristics of output files #O23 and #O26, and input file #I22:

```
>S #I12, #032,23
```

The display appears as follows:

```
#FILE    #JOB    FNAME    STATE   DEV/CL    PR  COP  RFN OWNER
#O23      #J6    $STDLIST READY   LP         2    1      MANAGER.SYS
#O32     #J444   SLP      READY   PP         2    1      FIELD.SUPPORT
#I12     #J491   $STDIN   OPEN    10                     MGR.STATS
#FILE    LDEV    LABEL      SECTORS         LINES        TIME
#O23      %1    %546357       48             96         22:15  9/17/87
#O32      %1    %1276156     1484           5075          2: 4  9/19/87
#I12      %14   %1033401        8             22
```

TEXT

Accesses an output spool file for use by FIND or LIST.

Syntax

>TEXT *dfid*

Parameters

dfid Device file ID, which identifies the output spool file, in the form #0*nnn*. #0 is optional.

Operation

The TEXT command accesses a READY output spool file, converts it to the LOCKED state, and permits you access to it using the COPY, APPEND, FIND, and LIST commands. If you text a file that is already accessed by TEXT, the file is closed and returned to the READY state. When you terminate SPOOK, any texted file is automatically closed. You can abbreviate the TEXT command as T.

Examples

To open spool file #022 for access by other SPOOK commands:

```
>TEXT #022
```

or

```
>T #022
```

or

```
>T 22
```

XPLAIN

Lists and describes the SPOOK command syntax.

Syntax

```
>XPLAIN
```

Operation

The XPLAIN command can be abbreviated as X.

The XPLAIN command displays this information:

```
EXIT  <<TERMINATE IF NOT A CHILD PROCESS>>
XPLAIN
SHOW   [ USER [ .ACCOUNT ] ] [ ; [@] [I] [O] ]
SHOW   DEVICEFILEID [ , DEVICEFILEID ]...
TEXT   DEVICEFILEID
LIST   [ RANGE ]
FIND   [ @ ] [ "STRING" ] [ , FRANGE ]
MODE   [ OPTION [ , OPTION ]...]
       OPTION = WIDTH / CONTROLS
ALTER {DFID [,DFID[,...]]} [ ; OPTION [ , OPTION ]...]
ALTER {USER [.ACCOUNT]   } [ ; OPTION [ , OPTION ]...]
       OPTION = PRI / COPIES / DEV
PURGE  DEVICEFILEID [ , DEVICEFILEID ]...
INPUT  [ USER [ .ACCOUNT ] ] ; TAPEFILE
INPUT  DEVICEFILEID [ , DEVICEFILEID ]... ; TAPEFILE
OUTPUT [ USER [ .ACCOUNT ] ] ; TAPEFILE [; PURGE]
OUTPUT DEVFILEID [, DEVFILEID ] ... ; TAPEFILE [; PURGE]
HELP
RUN    PROGRAMFILENAME [ .GROUP [ .ACCOUNT] ]
KILL  << CHILD PROCESS >>
QUIT  << TERMINATE >>
COPY   [RANGE] [,FILENAME]
COPY   [DFID [,DFID [,...]] ;] [RANGE [,FILENAME]]
COPY   [USER [.ACCOUNT] ;]  [RANGE [,FILENAME]]
APPEND [RANGE [,FILENAME]]
APPEND [DFID [,DFID [,...]] ;] [RANGE [,FILENAME]]
APPEND [USER [.ACCOUNT] ;]  [RANGE [,FILENAME]]
       [END  ]
```

SPOOK4/SPOOK5 Messages

SPOOK4/SPOOK5 displays warning and error messages as you use it. A warning message indicates an inability to initiate a requested command, or the truncation of some output. SPOOK warnings are in the following form, where *msgno* is a two-digit code for the message and *message* the message text:

```
*WARNING = msgno * message
```

An error message indicates illegal syntax or parameters, and occurs immediately after command entry. Error messages may also appear after displays produced by the SHOW, PURGE, INPUT, and OUTPUT commands. They indicate that an operation did not succeed on a certain spool file. Errors appear in the format:

```
[#FILE] *ERROR=errornum [BYTE=byteno] * message
```

#FILE appears only in error messages about output files; *errornum* is the two-digit error number; *byteno* identifies the character causing the error; and *message* is the text of the message.

Warning Messages

Error No.	Message	Action
1	NOT INTERACTIVE SESSION	SPOOK must be run interactively, not as a batch job.
2	END OF FILE	SPOOK encountered an end-of-file mark in your response, such as a colon (:) in column one. Run SPOOK again and respond correctly.
3	TOO MANY FILES	You have asked SPOOK to handle too many files in one operation. Reissue the command for fewer files.
4	INSUFFICIENT CAPABILITY	You lack the MPE capability, such as system manager (SM) or privileged mode (PM), to perform a command. Call your system manager, or try another command.

Error Messages

Error No.	Message	Action
19	IMPOSSIBLE INTERNAL ERROR	SPOOK system error. Call your HP system engineer.
20	INVALID COMMAND NAME	Re-enter the command with correct spelling and syntax.
21	COMMAND NAME TOO BIG	Re-enter the command with correct spelling and syntax.
22	PROMPT I/O ERROR	An input/output system transmission error occurred while SPOOK was sending its prompt to your terminal. SPOOK will try again.
23	INPUT I/O ERROR	An input/output system transmission error occurred while SPOOK was reading from the terminal. SPOOK will try again.
24	UNABLE TO CLOSE FILE	SPOOK cannot close a spool file; for example, it cannot return a current text file to the system after the execution of a new TEXT command. Re-enter command.
25	UNABLE TO PURGE FILE	SPOOK cannot delete the spool file referenced in the PURGE command. Type the PURGE command again.
26	FILE READ ERROR	SPOOK cannot access the text file. Use TEXT again, then retry your FIND or LIST command.
27	FILE FCONTROL ERROR	SPOOK cannot perform an input/output control operation, such as rewinding the tape. Retry your command.
28	FILE NOT 'READY'	You attempted to access a spool file not in the READY state. Wait until the state changes.
29	UNABLE TO OPEN FILE	To text in or purge a file, it must be in the READY state.
30	INPUT FILE NOT ALLOWED	You cannot access an input spool file in this context.
31	FILE NOT FOUND	The requested file does not exist. Check the device file ID and spelling.
32	INVALID FILE ID	The requested device file ID does not exist. Check your syntax, or use SHOW to verify the ID.
33	UNEXPECTED CHARACTER	Input syntax error. Try again.
34	USER NAME TOO BIG	User names must be eight characters or less. Check name and retry.
35	USER NOT ACCESSIBLE	The requested user name does not exist, or cannot be accessed by you. Check spelling.
36	ACCOUNT NAME TOO BIG	Account names must be eight characters or less. Check spelling.
37	ACCOUNT NOT ACCESSIBLE	The requested account name does not exist, or is inaccessible by you.
38	INVALID LINE MNEMONIC	Check your syntax and the spelling of FIRST and LAST in the range specification.

Error No.	Message	Action
39	INVALID LINE NUMBER	The specified line number does not exist in the file. Check syntax.
40	INVALID LINE COUNT	The count parameter is incorrect. Check the syntax.
41	INVALID LINE RANGE	Your range specification does not reference any lines in the file.
42	NON TERMINATED CHARACTER STRING	At least one quotation mark is missing. Check the syntax.
43	INVALID OPTION NAME	Illegal keyword in the MODE or ALTER command. Check the syntax and spelling.
44	INVALID OPTION SEPARATOR	The equals sign (=) is missing. Check the syntax.
45	INVALID OPTION PARAMETER	An illegal parameter follows the keyword. Check the syntax and spelling.
46	NO TEXT FILE	You must text in a file before using FIND or LIST.
47	FILE NOT 'READY/OPEN'	ALTER must be used on READY or OPEN files. You cannot alter a file that is LOCKED or ACTIVE.
48	TEXT FILE NOT ALLOWED	You may not access the current text file with this command.
49	MISSING SEMICOLON	Check the syntax.
50	UNABLE TO OPEN TAPE FILE	Be sure the tape is properly mounted and the tape unit is on line.
51	UNABLE TO CLOSE TAPE FILE	Dismount the tape manually. This automatically closes the tape file.
52	INVALID TAPE FILE	The file designator is bad. Check your file equations and be sure the file was correctly backreferenced with a "*".
53	INVALID TAPE FORMAT	You cannot input a file that was not previously output to this tape. Be sure the tape is correct.
54	TAPE FILE READ ERROR	Try again.
55	TAPE FILE WRITE ERROR	Try again. It is also possible that the spool file on the SPOOK tape is larger than the configured maximum spool file size on your system.
56	USER.ACCOUNT NOT ALLOWED	Do not reference files by creator names in this command. Use a device file ID.
57	NO EQUIVALENT DEVICE	This system does not have the device SPOOK is looking for, probably because the OUTPUT command was done on another system.
58	NO EQUIVALENT CLASS	This system does not have the device class SPOOK is looking for, probably because the OUTPUT command was done on another system.

Error No.	Message	Action
59	NO ROOM IN DEVICE TABLE	SPOOK cannot restore all the requested files at this time. Try again later.
60	MULTI VOLUME ABORT	You aborted a multi-volume INPUT command by responding N to: CHANGE VOLUMES ON LDEV *nn*? N If this action was a mistake, re-enter the INPUT or OUTPUT command.
69	INVALID LENGTH OF RECORD IN TEXT FILE	SPOOK cannot list the text file. Use the TEXT command again and retry the listing. If it fails again, recreate the spool file and purge the bad one.
70	FILE IS NOT PROGRAM FILE	The file does not exist, or is not a program file. To verify, type "LISTF *filename*,2".
71	NO CHILD PROCESS TO BE DELETED	Any child processes were already killed.
72	MISSING PROGRAM FILE NAME	Check the syntax.
73	UNABLE TO CLOSE COPY FILE (*nn*)	Error (*nn*) was returned by FCLOSE. Try again.
74	UNABLE TO OPEN COPY FILE (*nn*)	Error (*nn*) was returned by FOPEN. Refer to the description of the FCHECK intrinsic in the *MPE V Intrinsics Reference Manual* (32033-90007) for a list of error numbers.
75	SPOOLFILE CREATE ERROR	An attempt to open a new spool file failed.
76	UNABLE TO RENAME COPY FILE	The copy file has the same name as a permanent file.
77	DS COPY NOT YET AVAILABLE	This feature is not available.
78	LINE NUMBER IS IN PURGED EXTENT	You cannot list, append, or copy a line number in a purged extent.
79	INVALID COPY FILE	Illegal file name. Re-enter the correct file name or reference.
80	MISSING DFID OR USER.ACCOUNT	The device file ID or *username.accountname* must be supplied. Try again.

5

FREE2/FREE5

FREE2/FREE5 details the contiguous free space on each mounted disk volume, the total free space on each disk volume, and the total free space in the system. This utility aids in determining disk usage and the degree of disk space fragmentation. All values are decimal. No special capabilities are required to run FREE5.

The FREE5 utility (`FREE5.PUB.SYS`) runs on MPE V/E; the MPE V/P and MPE IV version is FREE2 (`FREE2.PUB.SYS`). The dialog and operation of both are identical. Throughout this chapter, FREE5 is used in all examples and explanations.

Operation

The output file for FREE5 is `$STDLIST` (the terminal). To run FREE5 with output directed to the terminal, type:

```
:RUN FREE5.PUB.SYS
```

The FREE5 output can be redirected by using the formal file designator `FREE5OUT`. To redirect the output to a line printer, use a file equation as follows:

```
:FILE FREE5OUT;DEV=LP
:RUN FREE5.PUB.SYS
```

To determine the amount of free space on a private volume, the private volume set must be logically mounted using the MPE `MOUNT` command:

```
:MOUNT vcsname.group.account
:RUN FREE5.PUB.SYS
```

A logical mount prevents a physical dismount of the private volume. Refer to the *MPE V Commands Reference Manual* (32033-90006) for a description of the `MOUNT` command.

Example

To run FREE5, type:

```
:RUN FREE5.PUB.SYS

FREE5 G.03.00 (C) HEWLETT-PACKARD CO.,1983
VOLUME  C11D1U0      LDEV 1     <---Volume name and logical device number.
LARGEST FREE AREA= 39848        <---Number of sectors in largest free area on
  SIZE  COUNT  SPACE   AVERAGE                this disk.
>100000 0      0       0
>10000  2      50104   25052
>1000   12     25062   2088
>100    28     8695    310      <---There are 28 free areas between 101 and
>10     122    4020    32         1000 sectors, with an average size of 310
>1      181    675     3          sectors, for a total of 8695 sectors.
TOTAL FREE SPACE=88556
*****************************
VOLUME  C11D1U1          LDEV 2
LARGEST FREE AREA= 10123
  SIZE  COUNT  SPACE   AVERAGE
>100000 0      0       0
>10000  1      10123   10123
>1000   12     23597   1966
>100    26     8363    321
>10     134    3239    24
>1      197    678     3          Total of SPACE column, which shows the
TOTAL FREE SPACE=46000            <---total free space on the volume.
*****************************
SYSTEM TOTAL FREE SPACE=134556    <---Total free space on all volumes.

END OF PROGRAM
:
```

FREE2/FREE5 Error Messages

You may receive any of the following messages when you use FREE2/FREE5.

Error Message	Action
LDEV #*nn* HAS #*mm* PAGES OF DISC FREE SPACE MAP MARKED AS BAD. UP TO #*pppp* SECTORS OF DISC SPACE MAY BE LOST.	The disk free space map (DFSM) is damaged. You must use COOLSTART to recover lost disk space.
ALLOCATION HAS BEEN DISABLED ON LDEV #*nn*.	Disk free space map (DFSM) is damaged. Existing files can be accessed, but new files cannot be created on this device.
LDEV #*nn* HAS BAD DFSM	You must use COOLSTART to recover lost disk space.
LDEV ##*nn* NOT MOUNTED, OR HAS NO DFSM.	If the logical device is mounted, the disk volume is in a pre-C0.00.08 free space format. Store the files on the disk, convert the disk to the new format using the VINIT subsystem's FORMAT and INIT commands, and restore the files.

6

MEMLOGAN

The memory logging analyzer (MEMLOGAN) lets the system manager print the error logs that were recorded by the memory error logging system.

MPE initiates MEMLOGP, the memory error logging process, when the system is initialized. Once an hour, MEMLOGP obtains error data from the memory error logging boards and writes the data to the two-record file MEMLOG.PUB.SYS. The time period may be adjusted with the MEMTIMER utility program, also described in this manual.

The memory logging facility is not related to the system logging facility that records events in the system log file.

Operation

Use the following procedures to use MEMLOGAN.

1. To run MEMLOGAN, type:

   ```
   :RUN MEMLOGAN.PUB.SYS [;PARM=n]
   ```

   If *n* equals:

0	The memory log file is printed but not cleared. This is the default value of *n*.
1	This option displays and clears the memory log file. Execute MEMLOGAN with this option immediately after bringing up the system for the first time after changing the memory size.
2	Displays the memory log file and deletes the file after the next =SHUTDOWN command. If you have error-correcting memory, a new MEMLOG is built when the system is restarted. Errors continue to be logged in MEMLOG between the execution of this option and the actual shutdown time.

2. MEMLOGAN identifies itself and begins to output one of the error log summaries. Example listings appear later in this chapter.
3. When MEMLOGAN has output the whole error log, it terminates with the message:

   ```
   END OF PROGRAM
   :
   ```

4. If MEMLOGAN completes with MEMLOG clear and in the "no-error" state, as it does after being run with ;PARM=1, it displays the following message before it terminates:

   ```
   * NO ENTRIES IN MEMLOG FILE *
   ```

MEMLOGAN Environment

The default output device for MEMLOGAN is the terminal, but you can redirect the output to another device by using the file equation that references the formal file designator OUT:

```
:FILE OUT; DEV=LP
:RUN MEMLOGAN.PUB.SYS
```

Memory Error Log Formats

The descriptions that follow show the data formats recorded by the memory error logging system. There are three versions: one for the Series 37, one for the Series 39/40/42/44/48/52/58 and one for the Series 64/68/70. The kinds of errors recorded are the same for all series of computers. The main difference in the formats is the presence of the controller field in the data for the Series 39/40/42/44/48/52/58.

MEMLOGAN on Series 37

Address		Error Type			Error
BOARD	ROW	TYPE	BIT	CHIP	COUNT
board	*row*	*type*	*bit*	*chip*	*count*

Field	Content
board	The memory module board on which the error occurred.
row	The row designation on the board in which the failing chip is located.
type	Type of error detected, as follows:
	`CHECK` Check bit error.
	`DATA` Data bit error.
	`MULTIPLE BIT ERROR` Error is more than one bit.
	`FORCED D.E.W.` Forced double error write. Parity error on the data transmitted to memory.
	`MISSING ARRAY BOARD` Non-responding array board.
bit	If *type* is `CHECK`, then *bit* is the failing check bit. If *type* is `DATA`, then *bit* is the failing data bit.
chip	The chip on which the error occurred, in the format `U`*n*, where *n* is a digit indicating the chip number.
count	The number of logging intervals during which this error was detected at least once. This value does not represent the number of times that an error was actually detected.

MEMLOGAN on Series 30/39/40/42/44/48/52/58

Address			Error Type			Error
Controller	Board	Row	Type	Bit	Chip	Count
controller	*board*	*row*	*type*	*bit*	*chip*	*count*

Field	Content
controller	The memory controller where the error occurred.
board	The memory module board on which the error occurred.
row	The row designation on the board in which the failing chip is located.
type	Type of error detected, as follows:
	`CHECK` Check bit error.
	`DATA` Data bit error.
	`MULTIPLE BIT ERROR` Error is more than one bit.
	`FORCED D.E.W.` Forced double error write. Parity error on the data transmitted to memory.
	`MISSING ARRAY BOARD` Non-responding array board.
bit	If *type* is `CHECK`, then *bit* is the failing check bit. If *type* is `DATA`, then *bit* refers to the failing data bit.
chip	The chip on which the error occurred, in the format `U`*n*, where *n* is a digit indicating the chip number.
count	The number of logging intervals during which this error was detected at least once. This value does not represent the number of times that an error was actually detected.

Interactive Input for Series 4X/5X Only

You are prompted for the number of each type of memory board installed, as follows:

```
HOW MANY 30173-60001 (2MB) BOARDS ARE INSTALLED? n1

HOW MANY 30478-60001 (2MB) BOARDS ARE INSTALLED? n2

HOW MANY 30479-60001 (4MB) BOARDS ARE INSTALLED? n3

HOW MANY 30161-60001 (1MB) BOARDS ARE INSTALLED? n4
```

MEMLOGAN on Series 64/68/70

Address		Error Type			Error
Board	Word	Type	Bit	Chip	Count
board	*word*	*type*	*bit*	*chip*	*count*

Field	Content
board	The memory module board on which the error occurred.
word	The word, within the data block, where the error occurred.
type	Type of error detected, as follows:
	`CHECK` Check bit error.
	`DATA` Data bit error.
	`MULTIPLE BIT ERROR` Error is more than one bit.
	`FORCED D.E.W.` Forced double error write. Parity error on the data transmitted to memory.
	`MISSING ARRAY BOARD` Non-responding array board.
bit	If *type* is `CHECK`, then *bit* is the failing check bit. If *type* is `DATA`, then *bit* is the failing data bit.
chip	The chip on which the error occurred, in the format `U`*n*, where *n* is a digit indicating the chip number.
count	The number of logging intervals during which the system detected this error at least once. This value does not represent the number of times that an error was actually detected.

Examples

If MEMLOG was updated with no errors found, MEMLOGAN prints the date and time of the first and last updates, and the logging interval at the time of the last update. The interval appears in the form of *hh:mm:ss*, indicating hours, minutes, and seconds. MEMLOGAN then terminates after displaying:

```
*** NO ERRORS LOGGED ***
```

If errors were logged, MEMLOGAN prints the date and time of the first and last logging, the date and time of the first and last logged errors, and an error table.

Series 37 MEMLOGAN Output

```
:RUN MEMLOGAN.PUB.SYS

MEMLOGAN G.03.00 (C) HEWLETT-PACKARD CO., 1980

LOGGING STARTED    - DATE:  5/25/84   TIME:  3:00
FIRST ERROR LOGGED - DATE:  6/28/84   TIME: 19:08
LAST ERROR LOGGED  - DATE: 11/19/84   TIME:  8:25
LAST LOG UPDATE    - DATE: 11/21/84   TIME: 12:39
TIMING INTERVAL    -  1:00:00


-----------------------------------------------
I    ADDRESS   I      ERROR TYPE      I ERROR I
-----------------------------------------------
I BOARD I ROW  I  TYPE   BIT   CHIP   I COUNT I
-----------------------------------------------
I   0   I  0   I  DATA     2   U1003  I     2 I
I   0   I  4   I  DATA    21   U606   I     1 I
I   0   I  9   I  CHECK    0   U101   I     1 I
I   0   I  3   I  DATA     8   U701   I    21 I
I   1   I  0   I  DATA     4   U1005  I     1 I
I   1   I  7   I  DATA    31   U308   I     3 I
-----------------------------------------------

END OF PROGRAM
:
```

Series 39/40/42/44 /48/52/58 MEMLOGAN Output

```
:RUN MEMLOGAN.PUB.SYS

HOW MANY 30173-60001 (2MB) BOARDS ARE INSTALLED? n1

HOW MANY 30478-60001 (2MB) BOARDS ARE INSTALLED? n2

HOW MANY 30479-60001 (4MB) BOARDS ARE INSTALLED? n3

HOW MANY 30161-60001 (1MB) BOARDS ARE INSTALLED? n4

MEMLOGAN G.03.00 (C) HEWLETT-PACKARD CO., 1980

LOGGING STARTED     -  DATE:  5/ 8/83   TIME: 15:16
FIRST ERROR LOGGED  -  DATE:  9/ 5/83   TIME:  9:18
LAST ERROR LOGGED   -  DATE:  8/29/83   TIME:  6:49
LAST LOG UPDATE     -  DATE: 10/12/83   TIME: 12:19
TIMING INTERVAL     -   1:00:00
```

ADDRESS				ERROR TYPE			
CONTROLLER	BOARD	ROW	TYPE	BIT	CHIP	ERROR COUNT
CONTROLLER A	0	1	CHECK	0	U198	2
		6	DATA	9	U103	13
	1	0	DATA	0	U19	4
		0	MULTIPLE BIT ERROR			1
		0	DATA	11	U149	3
	2	7	DATA	14	U172	2
	3	3	CHECK	5	U246	3

```
END OF PROGRAM
:
```

Series 64/68/70 MEMLOGAN Output

```
:RUN MEMLOGAN.PUB.SYS

MEMLOGAN G.03.00 (C) HEWLETT-PACKARD CO., 1980

LOGGING STARTED     - DATE:  6/ 6/83   TIME:  9:00
FIRST ERROR LOGGED - DATE:  9/15/83   TIME: 12:55
LAST ERROR LOGGED  - DATE:  3/28/84   TIME: 16:47
LAST LOG UPDATE     - DATE:  4/30/84   TIME: 14:05
TIMING INTERVAL     -  1:00:00


-----------------------------------------------
I    ADDRESS   I      ERROR TYPE      I ERROR I
-----------------------------------------------
I BOARD I WORD I  TYPE   BIT   CHIP   I COUNT I
-----------------------------------------------
I   0   I  0   I  DATA    10   U1202  I     1 I
I       I  2   I  DATA     6   U1606  I     1 I
I   1   I  1   I  DATA     2   U2004  I     1 I
I       I  2   I  CHECK    2   U 807  I     1 I
I       I  3   I  DATA     2   U2008  I     9 I
I   2   I  1   I  DATA    25   U1705  I     1 I
I   4   I  0   I  CHECK    5   U 503  I     1 I
I       I  1   I  DATA    24   U1805  I     1 I
I       I  3   I  DATA    11   U1108  I     1 I
I       I      I  DATA    16   U 608  I     1 I
-----------------------------------------------

END OF PROGRAM
:
```

MEMLOGAN Error Messages

MEMLOGAN displays a message and terminates when it encounters an error. If the MPE file system detects an error, it is explained by the file information display, and followed by one of these messages:

```
* OUT FILE ERROR *       **For output errors.**

* LOG FILE ERROR *       **For errors in reading the MEMLOG.**
```

For more details on the file information display, refer to the *MPE V Intrinsics Reference Manual* (32033-90007).

If the error was detected by MEMLOGAN, and not the operating system, MEMLOGAN displays this message:

```
* MEMLOGAN ERROR: errnum *
```

If *errnum* is 1, MEMLOGAN could not lock `MEMLOG` for exclusive use. If *errnum* is 2, MEMLOGAN could not unlock `MEMLOG` for access by other processes.

7

MEMTIMER

MEMTIMER sets the interval of time that elapses between updates of the memory log file MEMLOG.PUB.SYS by the system process MEMLOGP. You must have system manager (SM) capability to run MEMTIMER. If you try to run MEMTIMER without SM capability, the following message is displayed:

```
** PROGRAM REQUIRES SYS. MGR. CAPABILITY **
```

The default logging intervals of ten minutes for the Series 37 and one hour for the Series 39/40/42/44/48/52/58/64/68/70 are usually sufficient to provide an overview of system memory performance. MEMTIMER lets you request more frequent logging of memory data. The default interval is re-established every time the system is initialized. Therefore, you must run MEMTIMER after every system cold load if you wish to have a different interval.

Operation

Use this procedure to run MEMTIMER.

1. To run MEMTIMER, type:

   :RUN MEMTIMER.PUB.SYS; PARM=*n*

   The *n* denotes the new logging interval, in seconds, from 1 to 32767.

2. MEMTIMER identifies itself and reminds you of the capability that you must have to continue. If you have the correct capability, MEMTIMER causes logging to begin:

   ```
   MEMTIMER G.03.00 (C) HEWLETT-PACKARD CO., 1976
   ```

3. When the logging is complete, MEMTIMER resets the periodic logging to the new interval, and suspends itself with the message:

   ```
   END OF PROGRAM
   :
   ```

Example

To change the logging interval to ten seconds, enter:

```
:RUN MEMTIMER.PUB.SYS; PARM=10

MEMTIMER G.03.00 (C) HEWLETT-PACKARD CO., 1976
END OF PROGRAM
:
```

MEMTIMER Error Messages

You may receive these messages when you are using MEMTIMER.

Message	Action
INVALID PARM (DELAY) VALUE	You entered an invalid value for *n;* the value is out of range. The current interval remains in effect, and MEMTIMER terminates. Try again with the correct value.
MEMORY LOGGING PROCESS NOT ACTIVE	The logging program MEMLOGP was not active. (Perhaps a file system error caused MEMLOGP to abort.) Try MEMTIMER again.
MEMLOGP TIMER ENTRY NOT FOUND	Memory logging hardware is absent from system, or logging is currently in progress. If the hardware is installed, run MEMTIMER again to be sure MEMLOGP recognizes the new interval.

8

LISTLOG2/LISTLOG5

The program LISTLOG5 (LISTLOG5.PUB.SYS) runs on MPE V/E systems; the MPE V/P and MPE IV equivalent is LISTLOG2 (LISTLOG2.PUB.SYS). The dialog and operation of the two are identical. In this chapter, LISTLOG5 is used for all examples and discussion. You must have system manager (SM) capability to run LISTLOG2 and LISTLOG5. If you do not have SM capability, the following message is displayed:

```
** PROGRAM REQUIRES SYS. MGR. CAPABILITY **
```

LISTLOG5 analyzes files on the MPE system log. An MPE log file records events such as session or job initiation and termination, process termination, file closure, and system shutdown. Refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005) for more information on system logging.

Log files are named by the following convention, where *nnnn* is a four-digit number:

LOG*nnnn*.PUB.SYS

The formal file designator of the output file is LOGLIST, with the default device class LP. LOGLIST is opened as a new file and closed as a permanent file.

Operation

1. To find out which log files are on the system before you run LISTLOG5, enter the following:

   ```
   :LISTF LOG@.PUB.SYS
   ```

   MPE returns a list of numbers for all of the log files currently on the system. You can choose from the following numbers when you run LISTLOG5.

   ```
   FILENAME

   LOG          LOG1958     LOG1959     LOG1960     LOG1962
   ```

2. To run LISTLOG5, type:

   ```
   :RUN LISTLOG5.PUB.SYS
   ```

3. LISTLOG5 identifies itself and asks for the number of the first log file to print:

   ```
   LISTLOG5 G.03.01 (C) HEWLETT-PACKARD CO., 1982
   ENTER FIRST AND LAST LOG FILES TO BE ANALYZED
   FIRST? 1958
   ```

Look at the list of the log files displayed. Enter the four-digit number of the first log file you want printed. If you only want to analyze one file, enter it as the first file number and press (Return) in response to the LAST? prompt.

4. You are then prompted for the four-digit number of the last log file you want to print. Press (Return) to list only the first file, or enter the number of the last file; for example:

```
LAST? 1962
```

5. LISTLOG5 now displays a numbered list of events for which histories can be printed:

```
ENTER EVENTS TO BE PRINTED
TYPE NO.        EVENT
  0        LOG FAILURE
  1        SYSTEM UP
  2        JOB INITIATION
  3        JOB TERMINATION
  4        PROCESS TERMINATION
  5        FILE CLOSE
  6        SYSTEM SHUTDOWN
  7        POWER FAILURE
  8        SPOOLING LOG RECORD
  9        LINE DISCONNECTION
 10        LINE CLOSE
 11        I/O ERRORS
 12        VOLUME MOUNT/DISMOUNT
 13        VOLUME SET MOUNT/DISMOUNT
 14        TAPE LABELS
 15        CONSOLE LOG
 16        PROGRAM FILE EVENT
 17        CALL PROGRESS SIGNALS
 18        DCE PROVIDED INFO
 36        RESTORE
 37        PRINTER ACCESS FAILURE
 38        ACD CHANGES
 39        STREAM INITIATION
 40        USER LOGGING
 41        PROCESS INITIATION
 42        SECURITY CONF. CHANGES
 44        FILE OPEN
 45        COMMAND ACCESS
 46        MAINTENANCE LOGGING
 47        DIAGNOSTIC CONTROL UNIT
```

6. At the end of the list of events, you are prompted for input with the message:

```
ENTER EVENT NUMBERS SEPARATED BY COMMAS?
A CARRIAGE RETURN ASSUMES ALL EVENTS WILL BE EVALUATED
```

   Type the event numbers and press (Return).

7. LISTLOG5 then asks:

```
DO YOU WANT TO PURGE LOG FILES (Y/N)?
```

   If you answer YES, the log files are printed and then purged from the system. If you answer NO, the files are printed and retained by the system.

8. LISTLOG5 creates spool files of the events that you requested. There are no messages echoed back to your terminal if your request is successful. If your request is not successful, one of two messages is displayed: 1) an error message in the format described under "Error Conditions", or 2) the following message, indicating that there are no events in the log files whose numbers you requested:

```
NO DESIRED EVENTS FOUND IN LOG FILE nnnn
```

   In the example, events have been found for all requested log files except #1960 and #1962. Events in log files 1958 and 1959 are displayed, followed by these two messages:

```
NO DESIRED EVENTS FOUND IN LOG FILE 1960
NO DESIRED EVENTS FOUND IN LOG FILE 1962
```

9. LISTLOG5 now asks if you want to rerun the program; type YES or Y to continue with LISTLOG5, NO or N to terminate:

```
DO YOU WISH TO RUN AGAIN (Y OR N)? N
```

Examples

To print an MPE system log file, type:

```
:RUN LISTLOG5.PUB.SYS

LISTLOG5 G.03.01 (C) HEWLETT-PACKARD CO., 1982
ENTER FIRST AND LAST LOG FILE TO BE ANALYZED
FIRST? 2824
```

Enter a four-digit number.

```
LAST? 2825
```

Press Return if you only want to print file #2824.

```
        ENTER EVENTS TO BE PRINTED
            TYPE NO.    EVENT
             0      LOG FAILURE
             1      SYSTEM UP
             2      JOB INITIATION
             3      JOB TERMINATION
             4      PROCESS TERMINATION
             5      FILE CLOSE
             6      SYSTEM SHUTDOWN
             7      POWER FAILURE
             8      SPOOLING LOG RECORD
             9      LINE DISCONNECTION
            10      LINE CLOSE
            11      I/O ERRORS
            12      VOLUME MOUNT/DISMOUNT
            13      VOLUME SET MOUNT/DISMOUNT
            14      TAPE LABELS
            15      CONSOLE LOG
            16      PROGRAM FILE EVENT
            17      CALL PROGRESS SIGNALS
            18      DCE PROVIDED INFO
            36      RESTORE
            37      PRINTER ACCESS FAILURE
            38      ACD CHANGES
            39      STREAM INITIATION
            40      USER LOGGING
            41      PROCESS INITIATION
            42      SECURITY CONF. CHANGES
            44      FILE OPEN
            45      COMMAND ACCESS
            46      MAINTENANCE LOGGING
            47      DIAGNOSTIC CONTROL UNIT  Series 64 and 68 only.

    ENTER EVENT NUMBERS SEPARATED BY COMMAS?
    A CARRIAGE RETURN ASSUMES ALL EVENTS WILL BE EVALUATED.
    11
    DO YOU WANT TO PURGE LOG FILES (Y/N)? YES

    DO YOU WISH TO RUN AGAIN (Y OR N)? N
```

To redirect the LISTLOG5 output to disk, then copy the contents to a line printer (LP), enter the following:

```
:FILE LOGLIST=LOGFILE;DEV=DISC;REC=-132,1,F,ASCII;CCTL
:RUN LISTLOG5.PUB.SYS
    .
    .
    .

END OF PROGRAM
:FILE LP;DEV=LP;CCTL
:FCOPY FROM=LOGFILE;TO=*LP
    .
    .
    .
```

To redirect the LISTLOG5 output to tape, then copy it to a line printer (LP), enter the following:

```
:FILE LOGLIST=LOGTAPE;DEV=TAPE;REC-132,1,F,ASCII;&
CCTL;ACC=APPEND
```

The old output is then stored on tape.

```
:RUN LISTLOG5.PUB.SYS
    .
    .
    .
END OF PROGRAM

:FILE LP;DEV=LP;CCTL
:FILE LOGTAPE;DEV=TAPE;REC=-132,1,F,ASCII;CCTL
:FCOPY FROM=*LOGTAPE;TO=*LP
    .
    .
    .
```

Auditing User Activity

If job initiation logging is enabled, a system manager can audit the actions of one or more users based on individual identity. The **NAME** option of LISTLOG5 displays log records by named user. To access this feature, type:

```
:RUN LISTLOG5.PUB.SYS,NAME

LISTLOG5 G.03.01 (C) HEWLETT-PACKARD CO., 1982
ENTER FIRST AND LAST LOG FILE TO BE ANALYZED
FIRST? 2809
LAST? 2854
```

Press (Return) if you only want to print the first file.

```
ENTER EVENTS TO BE PRINTED

TYPE NO.    EVENT
  0     LOG FAILURE
  1     SYSTEM UP
  2     JOB INITIATION
  3     JOB TERMINATION
  4     PROCESS TERMINATION
  5     FILE CLOSE
  6     SYSTEM SHUTDOWN
  7     POWER FAILURE
  8     SPOOLING LOG RECORD
  9     LINE DISCONNECTION
 10     LINE CLOSE
 11     I/O ERRORS
 12     VOLUME MOUNT/DISMOUNT
 13     VOLUME SET MOUNT/DISMOUNT
 14     TAPE LABELS
 15     CONSOLE LOG
 16     PROGRAM FILE EVENT
 17     CALL PROGRESS SIGNALS
 18     DCE PROVIDED INFO
 36     RESTORE
 37     PRINTER ACCESS FAILURE
 38     ACD CHANGES
 39     STREAM INITIATION
```

```
40      USER LOGGING
41      PROCESS INITIATION
42      SECURITY CONF. CHANGES
44      FILE OPEN
45      COMMAND ACCESS
46      MAINTENANCE LOGGING
47      DIAGNOSTIC CONTROL UNIT  Series 64 and 68 only.

ENTER EVENT NUMBERS SEPARATED BY COMMAS?

A CARRIAGE RETURN ASSUMES ALL EVENTS WILL BE EVALUATED.
```

Press Return to list all events.

```
USER IDENTIFICATION? JOHN.PAYROLL ** Job/session name, user**
                                        ** name.account name**
```

User identification cannot exceed 70 characters. Valid user identification formats are:

- *job/session name,user name.account name*
- *user name.account name*
- *@,user name.account name*
- *@.account name*
- *@,@.account name*
- *job/session name,@.account name*

If *job/session name* is not included in the first format, LISTLOG5 interprets it as "*job/session name* = @", the functional equivalent of *@,user name.account name*. The formats *@.account name* and *@,@.account name* are also functionally equivalent. Both formats select all jobs/sessions of all users in the account *account name*.

Spaces before or after the delimiters are ignored. LISTLOG5 displays the following error message if you enter a user identification that contains a special character or numeric symbol:

```
*** ERROR, INVALID USER ID ***
```

After the error message is displayed, LISTLOG5 prompts again for user identification. After three unsuccessful attempts, LISTLOG5 asks:

```
DO YOU WANT TO PURGE LOG FILES (Y/N)? N
```

and then

```
DO YOU WISH TO RUN AGAIN (Y/N)? N
```

If you enter a valid user identification, LISTLOG5 displays this message:

```
DO YOU WANT TO PURGE LOG FILES (Y/N)?
```

It then lists the requested log event records relevant to the named user, as in the example below:

```
 USER     *GROUP     *ACCOUNT     *JSNAME    *INLDEV *JOB FILE NAME
 JOHN      DOE        PAYROLL      JREPORT     21      JTEST.GP.TACCT

*J/S NUM *LONUSER  *LONGROUP   *LONACCT  *JNAME  *SCHEDULED TIME
 #J112      QAMGR     QAGP        QA       JOBTST  WED,JAN 30,1989 10:00AM
```

The above record indicates that JOHN.PAYROLL streamed a job named JOBTST that was scheduled to run at 10:00 am Wednesday, January 30, 1989.

```
JOB NUM  *JOB NAME  *USER  *ACCOUNT  *DFID  *LDEV *FILE SIZE *STATUS
#J112     JREPORT    JOHN   PAYROLL  #0125  6      200         2
```

The above record indicates that JOHN.PAYROLL's job, JREPORT, spooled an output file #0125 with file size of 200 records to a printer whose logical device number was 6. The user had no WRITE access to the printer.

At the end of the report, LISTLOG5 displays a table of *job/session* numbers cross-referenced with *job/session name, user name* and *account,* as in the example below.

```
CROSS REFERENCE TABLE: J/S NUMBER   J/S NAME  USER   ACCOUNT
                       ----------   --------  ----   -------
                       #J112        JREPORT,  JOHN.  PAYROLL
                       #J50         JEST   ,  JOHN.  PAYROLL
                       #S130        STEST  ,  JOHN.  PAYROLL
```

LISTLOG2/LISTLOG5 Error Messages

On fatal errors, LISTLOG5 displays a message such as:

FOPEN ERROR TO LOG FILE *nnnn* ERROR NUMBER *yy*

where *nnnn* is the log file number, and *yy* is the FCHECK error code. For more information, refer to the *MPE V Intrinsics Reference Manual* (32033-90007).

After the FOPEN error message, LISTLOG5 prompts you with this message:

```
DO YOU WISH TO RUN AGAIN (Y OR N)?
```

If you answer YES or Y, LISTLOG5 starts over again with:

```
LISTLOG5 G.03.01 (C) HEWLETT-PACKARD CO., 1982
ENTER FIRST AND LAST LOG FILE TO BE ANALYZED.
```

9

ASOCTABL/ASOCTBL5

The ASOCTBL5 utility (ASOCTBL5.PUB.SYS) runs on MPE V/E systems; the MPE V/P and MPE IV version is ASOCTABL (ASOCTABL.PUB.SYS). The operation and dialog of both versions are identical. Throughout this chapter, ASOCTBL5 is used in all examples and explanations.

ASOCTBL5 is used by the system manager to create the device class/user association table in ASOCIATE.PUB.SYS. To use ASOCTBL5 you must have SAVE and WRITE access to PUB.SYS, or you must be logged on to the SYS account. This table defines which users are authorized to associate with which device classes. If you are authorized in the association table, you may use the ASSOCIATE command to gain access to a device class. By implication, this means that you can "associate" a device class such as LP to yourself for your exclusive use. Once gained, this association lasts until you log off or disassociate yourself.

In order for you to "associate" a device, no devices in this class can be previously associated by another user. After you "associate" a device class, you may execute any of the operator commands appropriate to this device. Messages for your associated device class appear on your $STDLIST device.

Only one user may associate to a given device class at any given time. If your device belongs to several device classes and one of these device classes has been associated to another user, you cannot use the ASSOCIATE command. The device is unavailable and you must be able to associate all devices in all of the classes that you are using.

Input to ASOCTBL5 is from a terminal or from a file. If ASOCTBL5 reads input from a file, the formal file designator is INPUT. ASOCTBL5 reads that file until it encounters an end-of-file marker, or until it finds a record starting with EXIT or exit in column 1. If no file equation exists, ASOCTBL5 prompts for an input from the terminal. While scanning the input, ASOCTBL5 builds a temporary file. When it has processed all of the input, it deletes the existing ASOCIATE file and saves the temporary file as ASOCIATE.PUB.SYS.

If errors are encountered in the input, ASOCTBL5 continues to scan the input following the error, but it does not delete the previous file or save the temporary file.

Operation

1. To run ASOCTBL5, type:

   ```
   :RUN ASOCTBL5.PUB.SYS
   ```

2. ASOCTBL5 identifies itself and prompts for an input.

   ```
   ASOCTBL5 G.03.00 (C) HEWLETT-PACKARD CO., 1979
   >
   ```

3. Enter a device class name followed by "=", and a list of user names and account names in the following form:

   ```
   > devclass = username.acctname [, ... ]
   ```

   For example:

   ```
   :LP=DANL.BOONE
   :TAPE=JOHN.BGOOD
   :EXIT
   ```

   The *devclass* must be a device class that exists in the current SYSDUMP configuration. When a device class is specified, all devices of that class are affected by the ASSOCIATE command.

   The *username* or *acctname* can be replaced by "@" to indicate all possible responses for that item:

user.@	Enables all users with the specified name in any account.
@.acctname	Enables all users in the specified account.
@.@	Enables all users.

   For example, to allow all users in the FINANCE account as well as TONY.ABC to associate all devices of class LP, enter the following:

   ```
   :LP=@.FINANCE,TONY.ABC
   ```

Examples

To list the users who are authorized to associate, and the device classes they may associate with, run the program in this way:

```
:RUN ASOCTBL5,LIST
```

The file ASOCIATE is not an editor file, and cannot be modified directly. You may, however, use an editor file as the input for ASOCTBL5 by using a file equation for the formal designator INPUT, as follows:

```
:EDITOR
/ADD
    1  LP=BBALL.JONES        ** Create the editor file **
    2  TAPE=FIELD.SUPPORT
    3  //
/K ASOT, UNN                 ** Keep the file unnumbered **
/E
END OF SUBSYSTEM

:FILE INPUT=ASOT              ** Establish the editor **
:RUN ASOCTBL5.PUB.SYS         ** file as the input file **

LP=BBALL.JONES
TAPE=FIELD.SUPPORT

END OF PROGRAM
:
```

If you choose not to use a file equation for INPUT, ASOCTBL5 prompts for input at the terminal, as follows:

```
:RUN ASOCTBL5.PUB.SYS
:LP=JIM.USERS
:EXIT

END OF PROGRAM
:
```

ASOCTABL/ASOCTBL5 Error Messages

All of these error messages denote fatal conditions. You can continue to make inputs after the appearance of one of these messages, but no modifications are made to the association table. You need to exit from the program and retry from the beginning.

```
UNABLE TO DELETE OLD 'ASOCIATE.PUB.SYS' FILE

EXPECTED AT LEAST 3 PARAMETERS, LDEV = USER.ACCT

= MUST FOLLOW LDEV

UNABLE TO OPEN INPUTFILE

CLASS NAMES ARE LIMITED TO 8 CHARACTERS

NO SUCH CLASS IN THIS SYSTEM

EXPECTED FOLLOWING USER NAME

UNABLE TO OPEN NEW ASOCIATE.PUB.SYS FILE
```

10

DPAN4/DPAN5

The DPAN4/DPAN5 program produces a formatted listing of main memory and virtual storage based on a memory dump taken after a system failure, HALT, or other abnormal shutdown. Your Hewlett-Packard application engineer can study the listing to obtain information about the failure. No special MPE capabilities are required to run DPAN4/DPAN5.

The DPAN5 utility (`DPAN5.PUB.SYS`) runs on MPE V/E; DPAN4 (`DPAN4.PUB.SYS`) runs on MPE V/P and earlier versions. The dialog and operation of both are identical. Throughout this chapter, DPAN5 is used in all examples and explanations.

If your system has more than 2 megabytes of memory, configure your spool files with the maximum number of sectors per extent (32764) so that they are big enough to hold all required data. In order to create a spool file this large, you may need to use the `>COND` command in the VINIT subsystem, as described in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

Note

Always save and label the original tape, cartridge tape, or serial disk generated in taking the dump. Your Hewlett-Packard application engineer may need this data, along with the DPAN5 printout.

Taking a Memory Dump

Before you can use the DPAN5 program to format the contents of main memory, you must dump this data to a reel tape, cartridge tape, or serial disk using the software dump facility, or SDF. Instructions are given for using the Series 37 computer, the Series 39/40/42/44/48/52/58 computers, and the Series 64/68/70 computers. If you need further information about the software dump facility, refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

Memory Dump on Series 37

Since the software dump facility stores main memory after a system failure, it operates in a stand-alone environment. For this reason, you should have one or more backup copies of the SDF on a serialized storage medium in addition to the copy you intend to use. Be prepared: you can make such backup copies only after the system is initially configured and is running. To ensure that you have the correct version of the software dump facility safely stored and ready to use at all times, create a new backup SDF each time you receive a software update.

For more information, refer to the discussion of "Backup Software Dump Procedures" in section X of the *MPE V System Operation and Resource Management Manual* (32033-90005).

Note

For a base system configuration consisting of a single HP 7906A disk drive operating in split mode (that is, the system is on the fixed disk, and the cartridge is used as a private volume/serial disk), the backup software dump is the only method of obtaining a main memory dump.

To obtain a memory dump, follow these steps:

1. Enter Ctrl B. If no READ is pending on the console, enter Ctrl A, then enter Ctrl B again. If you are still unable to get the Ctrl B prompt, call your Hewlett-Packard service representative.
2. The system displays the following message:

```
Welcome to Maintenance Mode

 2-LOCAL (from Normal)

 H for help-> DUMP

 Do you want to abort the system (Y/N)?Y

* MOUNT DUMP MEDIA, AND PLACE DRIVE ON-LINE.

* PRESS THE RETURN KEY TO CONTINUE EXECUTION OF SOFTDUMP.
```

Caution

You must serialize cartridge tapes before using them as dump media or the dump will fail. Refer to the VINIT subsystem in the *System Operation and Resource Management Reference Manual* (32033-90005) for instructions on serializing a cartridge tape.

3. The SDF program creates the dump, then initiates a WARMSTART, as follows:

```
DUMP
WARMSTART

HP32033G.XX.XX
WHICH OPTION <WARMSTART/COOLSTART>?WARM

DIRECTORY MAINTENANCE COMPLETED
PART 1 OF 6 COMPLETED - MEMORY RESIDENT TABLES SET UP
PART 2 OF 6 COMPLETED - SL BINDING
PART 3 OF 6 COMPLETED - SYSTEM I/O PROCESS CREATION
PART 4 OF 6 COMPLETED - DRIVER LOADING
PART 5 OF 6 COMPLETED - DISC RESIDENT TABLES SET UP
PART 6 OF 6 COMPLETED - SYSTEM PROCESS CREATION
WED, SEP 19, 1984,  7:22 PM? (Y/N) Y
```

```
LOG FILE NUMBER 793 ON
*WELCOME*
:HELLO OPERATOR.SYS;HIPRI
19:23/#S5/15/LOGON FOR: OPERATOR.SYS.PUB ON LDEV #20
:
```

4. If the system halts while loading the SDF, the following error message appears on the system console:

   ```
   ***DUMP FAILED
   ```

   This message indicates that an error has been detected, and the SDF is unable to execute. A backup software dump must be performed. Refer to "Creating a Backup SDF" in Section X of the *System Operation and Resource Management Reference Manual* (32033-90005) for step-by-step procedures.

   If the SDF command file does not exist or has been corrupted, the program automatically goes into interactive mode. The SDF prompts you for a command by printing **<SOFTDUMP COMMAND>**, at which point you enter the desired SDF command.

   If your serial storage medium is flawed, the following message may appear on the system console:

   ```
   SERIAL DEVICE I/O ERROR - You are now in
   SOFTDUMP INTERACTIVE MODE.  Please mount
   new media, then type "DUMP" in response
   to the prompt "<SOFTDUMP COMMAND>".
   ```

Notes

If you enter Ctrl Y from the system console at any time while the SDF file is executing, you may cause SDF to abort the current command and place you in interactive mode.

If you wish to alter your SDF command file, refer to "Changing the SDF Command File" in Section X of the *System Operation and Resource Management Reference Manual* (32033-90005).

If you want to print the dump to a line printer, run the DPAN4/DPAN5 utility program immediately after restarting the system.

Memory Dump on Series 39/40/42/44/48/52/58

Note

It is recommended that you have at least one backup copy of the software dump facility on serial disk, magnetic tape, or cartridge tape, regardless of your configuration. *A backup copy of SDF cannot be created after a software dump has failed.* Create the backup when the system is initially configured, and whenever you receive a software update. Refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005) for instructions on creating a backup copy of SDF.

To take a memory dump on Series 39/40/42/44/48/52/58, follow these steps:

1. Mount a serial disk, cartridge tape, or magnetic tape on the logical device specified by the device class `DDUMP` in the I/O configuration, then place the device on line.
2. On the system control panel, set the DUMP thumbwheel switch to the channel address and device address of the system disk drive.
3. Press the DUMP key on the system control panel, or enable the control and maintenance processor (CMP), by pressing Ctrl B. When you see the CMP prompt character (->), enter `DUMP` on the system console.
4. The software dump facility loads itself from the specified device and begins a serial execution of the SDF command file `SDFCOM`. The command file is located on the system disk and contains ASCII commands to do the following:
   - Prompt the operator to mount the dump medium and place the drive online.
   - Dump main memory.
   - Perform a `WARMSTART`.
5. If the SDF was loaded correctly, the following message appears at the system console:

   `SOFTWARE DUMP FACILITY (VER` *xx.xx/xx*`)`

   The system then executes the first instruction in the `SDFCOM` file.
6. Check to see that the serial storage medium is online, ready, and write enabled, and that the serial storage device is powered on. Press the Return key to continue execution of the SDF. For complete details on SDF commands, refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

Memory Dump on Series 64/68/70

Note

It is recommended that you create a backup copy of the software dump facility when the system is initially configured, and whenever you receive a software update. Then you have it when you need it. Refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005) for instructions.

To take a memory dump on Series 64/68/70, follow these steps:

1. Mount a magnetic tape on a logical device of device class DDUMP, then place the device on line.
2. If the C> or M> prompt does not appear on the console, press Ctrl B. When the prompt appears, type:

   DU[MP] [*i,c,d*]

   Where:

   i is the IMB number (0,1, or 2) of the system disk.

   c is the channel number (1-15) of the system disk.

   d is the device number (0-7) of the system disk.

   All values are decimal. Default values appear in the console banner. If you do not specify a new value for *i, c,* or *d,* then the previously specified value is used.

   The new values become the default values next time a dump is taken. If there is a power failure before the next dump, the factory set default values (0,2,1) take effect and may need to be corrected.
3. The SDF loads from the specified device and begins a serial execution of the SDF command file SDFCOM.
4. If the SDF loads correctly, the following message appears on the system console:

   * * * SOFTWARE DUMP FACILITY (VER *xx.xx/xx*)
5. Check to see that the serial storage medium is online, ready, and WRITE-enabled, and that the serial storage device is powered on. Then enter "RUN" in response to the DCU prompt on the console.

DPAN5 Dump Analysis Options

After you take a main memory dump using one of the preceding procedures, restart the system with a WARMSTART, COOLSTART, COLDSTART, UPDATE, or RELOAD.

Once the system has been restarted, four dump analysis options are available to you. Three of these options are performed in batch mode using the `STREAM` command. The fourth option is performed interactively. Your Hewlett-Packard application engineer can assist you in determining which analysis option to use. No `WRITE` ring is required on the tape.

Producing a DPAN5 Listing Using a Batch Job

The three options that produce a DPAN5 listing from a batch job are:

1. `STREAM DUMPJOB.PUB.SYS`. This generates a standard dump, a load map, and the I/O configuration.
2. `STREAM DUMPMINI.PUB.SYS`. This generates a minidump which includes all MPE formatted tables, the stack, and any segments associated with the processes which were current when the dump was taken, the load map, and the I/O configuration.

   The minidump reduces the amount of paper required to print a dump from a large memory system. However, it does not format all information available on the dump tape. For complete analysis of certain problems, it may be necessary to reprocess the dump tape to generate a standard dump.
3. `STREAM DUMPARCH.PUB.SYS`. This results in a dump tape, but produces no listing. Choose this option when you want to create a copy of the dump that can be transported to another system to be analyzed.

Producing a DPAN5 Listing Interactively

To obtain a DPAN5 listing interactively, use the DPAN5 `RUN` command. It has the following format:

`RUN DPAN5.PUB.SYS[,EIGHTLPI][,INFO=`*fname*`][;PARM=`*n*`]`

where

`EIGHTLPI` Format the dump with eight lines per inch.

fname Permanent dump file name (see text).

n When PARM=10 then interactive dialog is initiated.

Executing `DPAN5.PUB.SYS` produces a listing that reports on all data segments in memory, virtual memory, all MPE-resident data structures (formatted tables), the stack and associated segments of the current process at dump time, and other stacks of active processes in memory at the time of failure.

Running DPAN5 Without the INFO String

DPAN5 first attempts to open an input file named MDUMP in the logon group. If unsuccessful, DPAN5 attempts to open MDUMP.PUB.SYS. If also unsuccessful, DPAN5 opens a file whose formal file designator is MDUMP, with the default device class name TAPE. If you prefer serial disk over tape, set up a file equation before running DPAN5, with the formal designator of MDUMP; for example, FILE MDUMP;DEV=SDISC;REC=4096,,U. If you wish to use cartridge tape, the file equation must reflect the configured device class name of the cartridge tape.

Running DPAN5 With the INFO String

If you put a file name in the INFO string, you can use or create a permanent dump file by that name. This file is compatible with IDATE5. For example:

```
RUN DPAN5.PUB.SYS;INFO=dumpfile
```

If *dumpfile* exists, then DPAN5 checks to see if its MPE version is G.02.00 or later. If so, DPAN5 looks for and opens the accompanying virtual storage file, *Vdumpfile*. The dump is then processed.

If *dumpfile* does not exist, DPAN requests input (tape) and creates *dumpfile* as a permanent file. If the dump's MPE version is G.02.00 or later and virtual storage is present, *Vdumpfile* is also created. The dump is then processed.

DPAN5 directs its output to a file whose formal designator is DPANLIST, with device class name LP.

If you want a formatted listing of a few specific system tables, rather than a listing of the entire memory dump, you can initiate the interactive dialog with DPAN5 by entering the following:

```
RUN DPAN5.PUB.SYS;PARM=10
```

The system responds:

```
DPAN5 VER G.03.00 (C) HEWLETT-PACKARD CO. 1980
```

A tape request appears on the system console. When the tape is mounted, DPAN5 reads the dump, then prompts you for options. Levels of detail are indicated by the indentation. Reply YES to any prompt to get to the next level of detail. At the deepest level, reply YES to select that option. Reply NO to any prompt to reject the option and skip to the next prompt at the same level. Reply ALL to any prompt to select all options at deeper levels and skip to the next prompt at the same level. The prompt WHICH BANKS? requests a list of banks for DPAN5 to print.

DPAN5 asks:

```
MINIDUMP? YES or NO
```

If the minidump is invoked in this way, you can request dump data on additional data segments by data segment number. The responses to FORMAT TABLES? and PRINT PRIMARY MEMORY? supplement the minidump option.

For example, if the minidump option is specified, it is still possible to dump code segments in the PRINT PRIMARY MEMORY section. Or you may choose to dump only selected system tables, as specified by the responses to the FORMAT TABLES dialog. A sample dialog appears on the next page.

Respond to the first question in this example by entering a 4-digit DST number. You may use either decimal or octal, as long as the octal number is preceded by a % sign. Press (Return) to move to the next question level.

Respond to the remaining questions (except the last, WHICH BANKS?) by entering YES, NO, or ALL. ALL automatically specifies YES to all items in the next sublevel. Pressing (Return) is equivalent to selecting NO.

Respond to the last question, WHICH BANKS?, by entering decimal or octal number(s), separated by commas, or enter ALL.

```
WHICH DST (DECIMAL)?
    FORMAT TABLES?
       REGISTERS?
       PROCESS CONTROL?
          CST?
          DST?
          PCB?
          STACK MARKERS?
          INTERRUPT CONTROL STACK?
       MEMORY MANAGEMENT?
          CACHE DESCRIPTOR TABLE?
          SEGMENT LOCALITY LIST?
          VDS PAGE ALLOCATION?
          VIRTUAL DISK SPACE BIT MAP?
       I/O MANAGEMENT?
          DRT?
          INTERRUPT LINKAGE?
          LOGICAL PHYSICAL DEVICE?
          DIT?
          DISC REQUEST TABLE?
          IOQ?
          SYSTEM BUFFERS?
          TERMINAL BUFFERS?
       TIMER REQUEST LIST?
       SIR?
       MONITOR?
          DISC CACHING TABLES?
    PRINT PRIMARY MEMORY?
       FORMATTED?
          CODE SEGMENTS?
          CACHED REGIONS?
          FREE AREAS?
       WHICH BANKS?
```

At the end of this dialog, DPAN5 formats and prints the requested information.

Remove the tape from the drive. This comcludes the process of obtaining a memory dump.

Examples

Unless you requested specific information, the listing now contains:

- identification line
- register page
- code segment table (CST)
- code segment table extension (CSTX)
- data segment table (DST)
- process control block (PCB)
- monitor table
- segment locality lists
- available region lists
- memory region tables
- virtual disk space allocation information
- virtual disk space bit map
- device reference table (DRT)
- interrupt linkage table
- logical/physical device table (LPDT)
- device information tables (DIT)
- input/output request table
- disk request table
- system buffer information
- terminal buffer contents
- timer request list
- main memory contents (in octal)
- index
- formatted stack information
- formatted interrupt control stack (ICS)
- disk caching tables
- system internal resource table (SIR).

If you requested a minidump, the listing does not include the contents of main memory, but does include data on the current stack and segments. The example output shows some of the table headings you may expect to see in your output. The data begins with a Hewlett-Packard heading at the top of the page, followed by the boxed register table (if you answered YES to this question).

The last thing in the output is the dump index. What is between these two depends upon what you have requested, but can include any of the table headings listed in the following example:

```
HP3000 MEMORY DUMPG.00.00 OF SYS VER G UPDATE 00 FIX 00 DUMP TIME 3/14/84,5:40PM
(C) HEWLETT-PACKARD CO. 1980
                    ******    REGISTERS    ******
*************************************************************************
* DATA SEGMENT     * CODE SEGMENT   * STATUS=103113     * ISR=140017 SERIES 44 *
*************************************************************************
* DB BANK = 000012 * PB    = 144254 * MODE       = PRIV * RUN/HALT   = HALT    *
* DB      = 045230 * P     = 153226 * INTERRUPTS = OFF  * IRQ        = OFF   ***
* S BANK  = 000012 * PL    = 167637 * TRAPS      = OFF  * CSRQ       = OFF   ***
* DL      = 045100 * PBBANK= 000000 * STACK OP   = LEFT * PARITY     = OFF   ***
* Q       = 077426 * (P-PB)= 006752 * OVERFLOW   = OFF  * POWERFAIL  = OFF     *
* S       = 077436 *                * CARRY      = ON   * POWERON    = OFF     *
* Z       = 100363 *                * COND CODE  = CCE  * DISP FLAG  = OFF     *
*                  *                * SEGMENT #  = 113P * ICS FLAG   = OFF     *
*************************************************************************

             ******    FIXED LOW MEMORY    ******

 (ADDR %0)   CODE SEGMENT TABLE POINTER              006140

 (ADDR %1)   EXTENDED CODE SEGMENT TABLE POINTER     007724

 (ADDR %2)   DATA SEGMENT TABLE POINTER              002140


           ******    CST TABLE    ******

 SEGMENT                              REFERENCE            SEGMENT    ABSOLUTE ***
 NUMBER    SEGMENT NAME      MODE      BIT       TRACE    LENGTH     ADDRESS  ***
 -------   ------------      ----   ---------   -----    -------    --------


          ******    EXTENDED CST TABLE    ******

 SEGMENT  CSTBLK/PROCESS          REFERENCE            SEGMENT    ABSOLUTE ***
 NUMBER       INDX          MODE      BIT       TRACE    LENGTH     ADDRESS  ***
 -------  --------------    ----   ---------   -----    -------    --------


    ******    PROCESS CONTROL BLOCK (1ST HALF)    ******

    ******    PROCESS CONTROL BLOCK (2ND HALF)    ******

    ******    SIR TABLE    ******
```

```
******     DUMP INDEX    ******

NAME                                                    PAGE # FORMATTED        ***

 CODE SEGMENT TABLE                                          3
 DATA SEGMENT TABLE
 PROCESS CONTROL BLOCK                                       8
 CST EXTENSION                                               6
 SYSTEM GLOBAL AREA
 FIXED LOW CORE
 INTERRUPT CONTROL STACK
 SYSTEM BUFFERS                                             16
 UCOP REQUEST QUEUE
```

DPAN4/DPAN5 Error Messages

DPAN5 terminates after encountering an error. After any error, restream the job.

```
FWRITE failure in write to tape

FWRITE error temp file write failure

FREAD error in read from disk file

FCONTROL error in EOF write to tape

FSPACE error in movement of tape

FOPEN error - disk file open failed

FCLOSE error - temp file not closed

FCLOSE error - disk file not closed

File parity error - files invalid.
Dump may include several invalid file references in formatted table

Invalid dump tape - catastrophic tape errors - unable to read dump

Tape incorrectly prepared
```

11

SADUTIL

SADUTIL is a stand-alone program that performs emergency disk operations after the system has failed. It stores files on tape even if the file directory is corrupted. These files can then be reloaded onto the system with the RECOVER5 utility.

Caution

SADUTIL does not run under the safeguards of MPE, so careless use can corrupt the operating system. Use SADUTIL only in emergencies, or after the entire system has been backed up.

Because SADUTIL runs without operating system control, no special MPE capability is required to use it. It is generally used, though, by system managers and system supervisors.

SADUTIL is run after the system is shut down, and must be loaded from a tape prepared in advance. The procedure for preparing the tape is described in "Preparing a SADUTIL Coldload Medium" in this chapter. Once the system is halted, SADUTIL must be loaded and configured before it can be used. The steps are detailed in "Loading and Running SADUTIL". Once SADUTIL is running, the following commands are available:

`PDSK`	Prints an octal or ASCII dump of any given area of a specified disk volume.
`PDTT`	Prints the defective track table of a specified disk volume.
`PFIL`	Prints descriptions of files contained in the system file directory.
`PVOL`	Prints information contained in the volume label of a specified disk volume.
`EDIT`	Modifies the contents of a disk volume.
`FIND`	Searches a system disk for file labels.
`SAVE`	Retrieves files from disk and copies them to magnetic tape.
`COPY`	Copies the contents of one disk pack to another.
`OUTM`	Sets the output mode of print functions to printer output or console output.
`CLID`	Searches for the highest coldload ID, increments by one and establishes it as the new coldload ID. It also searches all `SYSDOMAIN` disks, and initiates the device configuration dialog for additional devices.
`CONF`	Initiates the device configuration dialog for additional devices.
`HELP`	Offers an explanation of all SADUTIL commands.
`STOP`	Terminates the SADUTIL program.

Preparing a SADUTIL Coldload Medium

Since SADUTIL recovers files after a system failure, it must be stored on an external medium before it is needed. To create the coldload medium, use the program COPYDUS. SADUTIL can be stored on a flexible disk, a reel tape, or a cartridge tape. If you use a flexible disk or cartridge tape, remember to format and serialize it before you run COPYDUS.

You must have the following files to create a copy of the diagnostic utility system (DUS) on flexible disk, magnetic tape, or cartridge tape:

```
COPYDUS.HP32231.SUPPORT     ** Program File **
FLOPDUS.HP32231.SUPPORT     ** Flexible Disk **
TAPEDUS.HP32231.SUPPORT     ** Magnetic Tape **
CARTDUS.HP32231.SUPPORT     ** Cartridge Tape **
```

1. To create a DUS medium, log on as follows:

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

2. Format and serialize your cartridge tape or flexible disk, using the FORMAT and SERIAL commands of the VINIT subsystem, as described in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

3. Check to be sure that you have the required files available, as follows:

   ```
   :LISTF @DUS.HP32231.SUPPORT
   ```

4. Run the COPYDUS program:

   ```
   :RUN COPYDUS
   ```

   COPYDUS identifies itself and asks for the identity of the medium on which you want to store the DUS:

   ```
   DUS COPY ROUTINE REVISION X.XX
   ENTER MEDIA TYPE (FLOPPY DISC, CARTRIDGE TAPE, MAG TAPE):MAG TAPE
   ```

5. Depending upon how you answer the MEDIA TYPE question, COPYDUS copies FLOPDUS, TAPEDUS, or CARTDUS onto your formatted and serialized scratch medium. You are reminded, before the copy starts, to have clean scratch medium:

   ```
   INSTALL SCRATCH TAPE.
   Reply to 'MTAPE' on the system console.
   ```

6. Mount the scratch medium at this time, either after the mount request, or before the mount reply.

7. The program identifies the medium you have mounted, and requests that you move to the system console to continue the dialog. When you respond to the tape request on the system console, the following message appears:

   ```
   BEGIN TRANSFER OF DATA.
   BEGIN VERIFICATION OF DATA.

   END OF PROGRAM
   :
   ```

Note

The latest version of SADUTIL is modified for MPE V/E. The MPE V/E version of the diagnostic utility system (DUS) tape functions correctly on an MPE IV or MPE V/P system, but the MPE IV and MPE V/P version does not work under MPE V/E. When you use SADUTIL, be sure that you have the correct version of the DUS tape.

Loading and Running SADUTIL

SADUTIL is loaded as a program file under the diagnostic utility system (DUS). Before using SADUTIL, you must coldload the DUS.

Coldloading the DUS on the Series 37

Follow these steps to coldload the DUS on the Series 37.

1. Mount the medium containing the diagnostic utility system (DUS) on the appropriate drive and place the drive online.
2. On the console, press Ctrl B to enter the Ctrl B mode of operation.
3. When the `H for help->` prompt is displayed, type `LOAD` and specify the channel and device on which the medium is mounted. This loads the DUS into memory. The following message is then displayed on the console:

```
DIAGNOSTIC/UTILITY SYSTEM    REVISION XX.XX

ENTER YOUR PROGRAM NAME      TYPE 'HELP' FOR PROGRAM INFO
:
```

4. Continue with the steps in "SADUTIL Configuration Dialog".

Coldloading the DUS on Series 39/40/42/44/48/52/58

Follow these steps to coldload the DUS on Series 39/40/42/44/48/52/58.

1. Mount the medium containing the diagnostic utility system on the appropriate drive and place the drive online, if applicable.
2. On the system control panel, set the LOAD thumbwheel switches to the channel address and device address of the drive containing the DUS.
3. You may load the DUS into memory from either the system control panel or the control maintenance panel (CMP):
   - From the system control panel, press the HALT key, then press the LOAD key.
   - From the CMP, enter Ctrl B to generate the CMP prompt (->). When it appears on the system console, type `HALT`. When another prompt is printed, type `LOAD`.

This loads the DUS into memory. The following message is then displayed on the console:

```
DIAGNOSTIC/UTILITY SYSTEM     REVISION XX.XX

ENTER YOUR PROGRAM NAME
:
```

4. Continue with the steps in "SADUTIL Configuration Dialog".

Coldloading the DUS on Series 64/68/70

Follow these steps to coldload the DUS on Series 64/68/70.

1. Mount the tape containing the DUS on the tape drive and place the drive online.
2. If the DCU prompt (C> or M>) is not already displayed on the system console, press Ctrl B. When you see the prompt, type HALT. When another prompt appears, type LOAD.

   If you wish to coldload from a device other than the preset default device, enter the IMB number, channel number and the device number, separated by commas, after the LOAD command, as follows:

   ```
   M> LOAD imb,channel,device
   ```

Note

Once you have overridden the default coldload device in this way, the new values become the default until power is cycled on the system. At that time, the default values initially set by the factory take effect.

The DUS is now loaded into memory and the following message appears on the console:

```
DIAGNOSTIC/UTILITY SYSTEM     REVISION XX.XX

ENTER YOUR PROGRAM NAME
:
```

3. Continue with the steps in the "SADUTIL Configuration Dialog".

SADUTIL Configuration Dialog

When you are prompted for a program name, respond as follows:

```
ENTER YOUR PROGRAM NAME
SADUTIL
```

SADUTIL identifies itself and begins the interactive configuration dialog:

```
DISC UTILITY X.XX.XX (C) HEWLETT-PACKARD CO., 1982
```

Note

You must enter all SADUTIL commands in uppercase letters.

SADUTIL requires you to specify the configuration of any disk on which an operation is to be performed. Usually, this means all system disks configured when the system failed. You must assign *ldev* 1 to the disk that MPE uses as "LDEV 1". Assign *ldev* 2 through *ldev n* to other disks, regardless of their MPE logical device designations.

You may use SADUTIL with private volumes or system volumes, but not both at the same time. To use it with private volumes, configure the master disk of the private volume as "LDEV 1" and all slave volumes as "LDEV 2", "LDEV 3", and so on.

1. `LIST LOGICAL DEVICES? Y`

   To print a listing of all the logical devices, DRT numbers, unit numbers, types, and subtypes, currently in the LDEV table, enter `YES` or `Y`. To skip the listing, enter `NO` or `N` or press Return.

   The printer configuration is part of the diagnostic utility system (DUS). To access the DUS, use the `STOP` command to exit SADUTIL (thereby entering MANAGER mode). Execute the `LISTIO` command, then enter

   `CHANGEIO` *printer* `TO` *chnl,dev*

   Finally, type `EXIT` to leave MANAGER mode and resume SADUTIL.

2. `DISC CONFIGURATION CHANGES? Y`

   To change or add devices to the logical device table, enter `YES` or `Y`. To skip and the leave table unchanged, enter `NO` or `N` or press Return.

3. `PRIVATE VOLUME SET? Y`

   To specify that you will be working with private volumes, enter `Y`.

4. `LOGICAL DEVICE? 1`

   Configure the master disk of the private volumes as *ldev* 1. Configure any slave private volumes as *ldev* 2, *ldev* 3, and so on. Then SADUTIL operates on the private volumes and not on the system volumes. To add, delete, or change the configuration, enter the logical device number (decimal). Press Return to go straight to step 9.

5. `DRT? 49`

   Enter the hardware DRT number (decimal) of the disk to be referenced by this *ldev*. Enter 0 to delete this *ldev* and return to step 4.

6. `UNIT? 0`

   Enter the hardware unit number (decimal) of the disk drive. This must be 0 for any type other than type 0 or 3 disks.

7. `TYPE? 3`

   Select the disk type from the "SADUTIL Device Types and Subtypes" list, and enter it.

8. SUB-TYPE? 8

   Enter the subtype. Refer again to the "SADUTIL Device Types and Subtypes" descriptions. Be sure that the type and subtype entries are known to the MPE operating system: an HP 7906 disk may need to be assigned more than one *ldev* to reference its various logical parts.

   The program continues to loop from here back to step 4 until all disks in the current volume set are configured, and you press (Return) at step 4.

9. LIST LOGICAL DEVICES? Y

   Enter Y or YES, or N or NO, or (Return). This prints a listing of all logical devices, DRT numbers, unit numbers, types, and subtypes currently in the LDEV table. *ldev* 1, the system disk (or master disk, if you are using private volumes), must be configured by this point. If it is not, SADUTIL takes you back to step 1.

10. SERIAL DEVICE CHANGES? Y

    To modify the configuration of the magnetic tapes or serial disks used by the SAVE and FIND commands, enter Y or YES. To leave it unchanged and skip to step 16, enter N or NO or (Return).

11. DRT? 41

    Enter the hardware DRT number (decimal) of the new device.

12. UNIT? 0

    Enter the unit number of the new device.

13. TYPE? 24

    Configure any tape or serial disk that is indicated as serial in the SADUTIL device types and subtypes description.

14. SUB-TYPE? 0

15. LIST SERIAL DEVICE? Y

    To list the serial device configured in steps 10 to 13, enter Y.

16. ENTER FUNCTION?

    This ends the configuration dialog. Enter one of the functions described on the following pages, or STOP to terminate SADUTIL. When you type STOP, control returns to the DUS.

SADUTIL Device Types and Subtypes

This summary consists of devices, types and subtypes recognized by the MPE operating system. Be sure that you use matching types and subtypes to describe your devices.

Device	Type	Subtype	Serial	Description
HP 7920A/S	0	8	*	Moving head disk
HP 7925A/S	0	9	*	Moving head disk
HP 7933A	3	8	*	Moving head disk
HP 7935H	3	8	*	Moving head disk
HP 7906A	0	10	*	Removable cartridge only
		11		Fixed platter only
		12		Entire drive
HP 9895	2	0	*	Flexible disk unit (single)
		1	*	Flexible disk unit (double)
HP 9140A	3	0	*	Cartridge tape unit
HP 9144A	3	3	*	Cartridge tape unit
HP 7911A	3	1		Winchester disk
HP 7912A	3	2		Winchester disk
HP 7914A	3	4		Winchester disk
HP 7945A	3	5	*	Winchester disk
HP 7970E	24	0	*	Magnetic tape drive
HP 7974A	24	3	*	800/1600 BPI magnetic tape drive
HP 7976A	24	1	*	1600/6250 BPI magnetic tape drive
HP 7978A	24	2	*	1600/6250 BPI magnetic tape drive

CLID

Searches for the highest coldload ID, increments by one and establishes it as a new coldload ID. Searches all SYSDOMAIN disks. Initiates the device configuration dialog for additional devices.

Syntax

```
CLID
```

Operation

If INITIAL is aborted during a COOLSTART or WARMSTART, and some coldload IDs have been updated and others have not, the next attempt to use INITIAL causes the system to report that the volume table is corrupt and that a RELOAD is necessary. Since the actual problem is that the coldload IDs do not match, you can use the CLID command to reset them to 1, and thus avoid performing the RELOAD.

Example

```
ENTER FUNCTION:  CLID

WARNING!! This function will rewrite all coldload ID's
Are all system domain volumes mounted and ready? Y

Coldload ID's written in system tables
Coldload ID written on LDEV#1
Coldload ID written on LDEV#2

ENTER FUNCTION:
```

CONF

Initiates the device configuration dialog for the configuration of additional devices.

Syntax

CONF [*ldev*]

Parameters

ldev A logical device number to be configured. Omit this parameter if you wish to be prompted for the device(s) to configure.

Operation

If you forgot to configure a device during the configuration dialog, use **CONF** to return to that phase now. Refer to the "SADUTIL Configuration Dialog" earlier in this section.

If *ldev* is specified, only that device is configured, and the dialog begins at step 5. If no *ldev* is specified, the dialog starts at step 2, and you can configure any device number.

Example

To configure *ldev* 6 (after the initial SADUTIL configuration dialog has been completed), proceed as follows:

```
ENTER FUNCTION:  CONF 6
DRT? 32
UNIT? 0
TYPE? 0
SUBTYPE? 10
LIST LOGICAL DEVICES? Y

 LDEV  DRT  UNIT  TYPE  SUBTYPE
 ----  ---  ----  ----  -------
   1    15    0     0      11
   2    15    0     0      10
   6    32    0     0      10

ENTER FUNCTION:
```

COPY

Copies the contents of one disk pack to another.

Syntax

COPY [*ldev*]

Operation

Note The COPY function is not a supported method of system backup.

If disks are of different types, the size of the smaller disk determines the amount of data copied. Both disk packs must have defective tracks reassigned. Packs with deleted tracks are not copied.

Disks of different types always have the defective track table copied. If the disks are the same type, the defective track table is handled as follows:

- If the "from" disk has not been initialized by MPE, the defective track table is copied to the "to" disk.
- If the "from" disk was initialized by MPE and the "to" disk was not, the defective track table is copied but all entries are deleted.
- If both the "from" and "to" disks are MPE-initialized, the defective track table is not copied.

The COPY function informs you if the disk is not MPE-initialized, or if the volume table information disagrees with the way you have configured the disk, by issuing the following messages:

```
*WARNING*   LDEV #nn NOT INITIALIZED
```

or

```
*WARNING*   LDEV #nn CONFIGURED SUBTYPE DOES NOT AGREE WITH VOLUME
TABLE
```

You are given the option of continuing after a warning.

Example

```
ENTER FUNCTION: COPY
FROM LOGICAL DEVICE? 1
TO LOGICAL DEVICE? 2
MOUNT SCRATCH PACK ON LDEV #2
Press <RETURN> when mounted.
ENTER FUNCTION:
```

EDIT

Modifies the contents of a disk volume.

Syntax

```
EDIT
```

Operation

This function prompts you with >. You may then request a disk modification by typing an EDIT command as described below. If you respond to the prompt by pressing (Return), SADUTIL exits from the EDIT function and prompts you for another function.

```
>OUTM {C}
      {P}
```

Sets the output mode for PDSK to the console (C) or the line printer (P).

>DISC [*ldev*]

Specifies the logical device on which the volume to be edited exists. (The initial *ldev* is 1).

>BASE [*basesector*]

Specifies the sector address to which all disk address references in the PDSK and MODIFY commands are relative. The default is 0.

>PDSK

Prints an octal/ASCII dump of any given area on the volume specified by the DISC command. (Same as PDSK function.)

>MODIFY [*diskaddr*] [,*wordloc*] [,*wordcount*]

Lets you modify specified words on the disk volume. Omitting a parameter is equivalent to specifying 0. You may use octal numbers for the parameters if you precede the number with a "%" sign. SADUTIL responds to MODIFY by displaying:

SECTOR % *sectoraddr* *wordloc* : *oldcontents* =

where:

sectoraddr is the absolute octal sector address, not relative to *basesector*.

wordloc is the octal location, within the sector, of the word to be modified.

oldcontents is the current octal value of the word to be modified.

The new contents of the word should be entered as a string of six or fewer octal characters. Entering / terminates MODIFY; you are then prompted for a new SADUTIL command.

For each word, EDIT displays this information, and confirms each modification by displaying WRITTEN.

Examples

To modify disk 2, sector 0 (relative to *baseaddr*), word 7, with printer output, enter:

```
 ENTER FUNCTION: EDIT
DISC 2
OUTM P
BASE 0
MODIFY 0,7,1
SECTOR % 0
 7: 012451:= (RETURN)
WRITTEN
```

In the next example, three words of sector 2 (relative to *baseaddr*) are modified, beginning at word 2. The slash mark (/) terminates MODIFY.

```
MODIFY 2,2,3
 SECTOR % 1
  2: 000014:=000014
  3: 000004:=/
WRITTEN
```

To modify one word of sector 1 (relative to *baseaddr*) beginning at location 2, enter:

```
MODIFY 1,2
 SECTOR % 1
   2: 000015:=14
WRITTEN
```

In the next example, three words of sector 0 (relative to *baseaddr*) are modified, beginning at location 126. Note two things in this example. One, the word is changed only if the new value differs from the existing data, and two, since there are 128 words per sector, the third word is in SECTOR % 1.

```
MODIFY 0,126,3
 SECTOR % 0

 176: 000177:= 0
 177: 000000:= 0
 SECTOR % 1
   0: 000003:= 2
WRITTEN
```

FIND

Scans a system disk for file labels.

Syntax

```
FIND
```

Operation

FIND allows files to be saved even when the system directory has been destroyed.

FIND operates very slowly; it is for emergencies only. The function expects file sets to be in the form *filename.groupname.accountname.* Any of those parameters can be replaced by the wildcard character "@", to specify all members of the set. This feature saves time when you are attempting to find several files.

Upon finding a file label, SADUTIL displays:

- The file name, group and account.
- The device number of the disk.
- The disk address of the file label.

Example

```
ENTER FUNCTION: FIND
FIND SCANS FOR FILE LABELS
THERE IS NO GUARANTEE THAT THE LABELS OR FILES ARE INTACT
IF THE FILES ARE REQUESTED TO BE SAVED, MANY MAY BE BAD OR ALREADY PURGED
IF THE FILE INFO ENDS WITH '?@%%%%', THEN THE FILE IS EITHER ALREADY
  PURGED OR IT WAS LEFT OPEN WHEN THE SYSTEM FAILED
FILES LISTED WITH '?@%%%%' WILL NOT BE SAVED
AN ATTEMPT CAN BE MADE TO SAVE VIA SAVE BY LDEV # AND SECTOR ADDRESS

DO YOU WISH FILES FOUND TO BE SAVED? Y

ENTER LDEV NUMBER TO SCAN: 1
ENTER FILE SET TO FIND: @.RANDAZZO.MPEM   ** File subset specific.**
   DATE? 1/1/83
SCANNING LDEV 1 FOR FILE LABELS

HIT CONTROL A OR Y OR BREAK TO STOP 'FIND'

PLACE SERIAL DEVICE ON LINE AND PREPARE IT FOR WRITE
SECTORS/TRACK=48
  TOTAL TRACKS=4075
```

If you answer Y to the question DO YOU WISH FILES FOUND TO BE SAVED?, FIND invokes the SAVE command. If you respond NO, you must save each file that you want to keep with the SAVE command. FIND produces this listing on the device selected by OUTM:

```
SCANNING LDEV 1 FOR FILE LABELS
  MODIFIED SINCE 1/ 1/83
  LDEV=  1,DRT=  4,UNIT=  0,TYPE=  0,SUBTYPE=  8

  FILENAME               CREATED  MODIFIED    ACCESSED     SECTOR

SL    .PUB    .SYS       11/ 1/72   5/ 3/83     8/ 3/83  %  15675 ?@%%%%
     SL       .PUB  .SYS          WAS EITHER PURGED OR LEFT OPEN

HIOTAPEO.PUB    .SYS      8/ 3/81   6/ 6/83     8/ 1/83  %  34417
     HIOTAPEO.PUB  .SYS             WAS SAVED!

 CSDUMMY.PUB    .SYS      9/23/81   6/ 6/83     7/ 1/83  %  34436
     CSDUMMY .PUB  .SYS          WAS SAVED!

IOCDRDO.PUB     .SYS      6/22/81   6/ 6/83     6/ 1/83  %  34444
     IOCDRDO .PUB  .SYS          WAS SAVED!
```

HELP

Offers an explanation of all SADUTIL commands.

Syntax

```
HELP
```

Example

```
ENTER FUNCTION: HELP

  Below are all the commands supported by SADUTIL

CLID                    Set all coldload IDs to 1
CONF [ldev]             Configure logical devices
COPY                    Copy one disk to another
DBUG                    Enter symbolic debugger
EDIT                    Below are the five edit commands
  BASE [basesector]     Specifies base sector number
  DISC [ldev]           Specifies logical device to edit
  MODIFY [diskaddr]
         [,wordloc]
         [,wordcount]   Modify a sector
  PDSK                  Print disk. Enter address as
                        [address][,sectcount][;A|O]
  OUTM [C|P]            Output to console or printer
FIND                    Scan a disk for file labels
HELP                    Explain facility
OUTM [C|P]              Output to console or printer
PDTT [ldev]             Print defective tracks table
PDSK [ldev]             Print disk. Enter address as
                        [address][,sectcount][;A|O]
PFIL                    Print file names from directory
PVOL [ldev]             Print volume label of ldev
SAVE                    Save files to serial device
STOP                    Exit SADUTIL

ENTER FUNCTION:
```

OUTM

Sets the output mode of print functions to printer or console.

Syntax

```
OUTM [C]
     [P]
```

Parameters

C	Directs output of print functions to the console.
P	Directs output to the line printer in 132-character lines.

Operation

The default output mode is C (the console).

Example

To set the output mode to the line printer, enter:

```
ENTER FUNCTION: OUTM P
```

To set the output mode to the console, enter:

```
ENTER FUNCTION: OUTM
```

or

```
ENTER FUNCTION: OUTM C
```

PDSK

Prints an octal or ASCII dump of any given area of a specified disk volume.

Syntax

PDSK [*ldev*]

Parameters

ldev The logical device number of the disk volume.

Operation

PDSK repeatedly prompts you with:

```
ENTER ADDRESS:
```

You may respond with the following information:

firstsect {,*numsect* } {;A[,O]} {:*lastsect* } {;O[,A]}

where:

firstsect is the starting sector number to dump.

numsect is the number of sectors to be dumped, starting with *firstsect.*

lastsect is the last sector number to dump. Sector numbers *firstsect* through *lastsect* will be dumped.

A prints the dump in ASCII.

O prints the dump in octal.

If you omit the A and O parameters, the dump is printed in whatever format you specified previously. The initial format is octal.

After the dump is printed, you are prompted again with ENTER ADDRESS:.

To abort the dump, press Ctrl Y. After aborting, PDSK asks you to enter another address. Press Return to terminate PDSK.

Example

To print an octal dump of sector 0 of the volume residing on logical device 1:

```
ENTER FUNCTION: PDSK
 ENTER ADDRESS: 0
 ENTER ADDRESS: RETURN
ENTER FUNCTION:
```

To print an ASCII dump of sectors 0 through 1 of the volume residing on logical device 1:

```
ENTER FUNCTION: PDSK 1
 ENTER ADDRESS: 0:1;A
 ENTER ADDRESS: RETURN
ENTER FUNCTION:
```

PDTT

Prints the defective track table of a specified disk volume.

Syntax

PDTT [*ldev*]

Parameters

ldev The logical device number of the disk volume.

Operation

MPE records defective disk areas in the defective track table of each volume. The HP 7911, 7912, 7914, 7933, 7935, and 7945 disk drives do not have defective track tables; do not use PDTT with them. The DTT is found in sector 1 (the second sector) in this format:

Word	Contents
0	Number of entries (n) in the table (ranges from 0 to 120).
1 - n	(Bits 0:14) = Track number. (Bits 14:2) = Status, where: 0 = Suspect track. 1 = Suspect alternate track. 2 = Deleted track. 3 = Reassigned track.
121-125	Reserved (filled with zeros).
126	Next available alternate track (moving-head disks only).
127	Logical disk pack size (in cylinders for moving-head disks, or tracks for fixed-head disks).

Example

To print the defective track table of the volume residing on logical device 1:

```
ENTER FUNCTION: PDTT 1
LDEV=   1, DRT=  15, UNIT=   0, TYPE=   0, SUBTYPE=   5
 LOGICAL PACK SIZE = 400 CYLINDERS, 11 ALTERNATE TRACKS AVAILABLE
 NO ENTRIES IN DTT
ENTER FUNCTION:
```

PFIL

Prints descriptions of files contained in the system file directory.

Syntax

```
PFIL
```

Operation

PFIL repeatedly prompts:

```
ENTER NAME:
```

User response is:

filename[.*groupname*[.*acctname*]] [,*detail*]

The parameters *filename, groupname,* and *acctname* can be replaced with "@" to signify *all members of the set.* Default is all files in system.

The parameter *detail* can be 0 (default), 1, or 2, depending on the type of information desired:

0	Requests a display of file names only.
1	Requests a display of the name, LDEV and sector address of the first extent (that is, the file label) in the designated group and the account.
2	Requests a display of the name, creation date, last modification date, and last access date of a file.

To abort the PFIL, press Ctrl Y.

If SADUTIL encounters an invalid file label during a PFIL function, it prints an asterisk (*) immediately after the file name.

Examples

To determine if the file SPL.PUB.SYS exists on the system:

```
ENTER FUNCTION: PFIL
 ENTER NAME: SPL.PUB.SYS

 ACCOUNT = SYS      GROUP = PUB

 SPL
```

To determine the date of creation, last modification, and last access for the file SPL.PUB.SYS :

```
ENTER FUNCTION: PFIL
 ENTER NAME: SPL.PUB.SYS,2

 ACCOUNT = SYS      GROUP = PUB

 SPL           8/29/74   12/30/75   1/29/76
```

To print all files in the SYS account:

```
ENTER FUNCTION: PFIL
 ENTER NAME: @.@.SYS
```

To print the file names, logical device numbers, and beginning sector address for all the files:

```
ENTER FUNCTION: PFIL
 ENTER NAME: @,1
```

PVOL

Prints information contained in the volume label of a specified disk volume.

Syntax

PVOL [*ldev*]

Parameters

ldev The logical device number of the disk volume.

Operation

The volume label, located in the first sector (sector 0) of the disk, is written in this format:

Words	Contents
0-5	On the system disk, this field contains the bootstrap input/output program. On other disks, this field is filled with zeros.
6	(Bits 6:6) Disk type. (Bits 12:4) Disk subtype.
7	Coldload count, incremented each time the system is coldloaded.
8-9	The characters 3000, used to verify that the disk label is valid.
10-13	The volume name, left-justified and padded with blanks.
14-127	Reserved (initialized to zeros).

Example

To print volume label information for *ldev* 2:

```
ENTER FUNCTION: PVOL 2
 LDEV=   2, DRT=  15, UNIT=   0, TYPE=   0, SUBTYPE=   4
 TYPE=   0, SUBTYPE=    4, C-L ID= 5419 , VOL. ID= SSTV2
```

SAVE

Retrieves files from disk and copies them to magnetic tape.

Syntax

```
SAVE
```

Operation

SAVE is used with the EDIT function (for editing disks) and the RECOVER5 program (for creating disk files) after a catastrophic system failure.

Note

SAVE copies any existing ACDs along with the files it copies. Refer to chapter 12 on RECOVER2/RECOVER5 for more information about restoring ACDs.

SAVE retrieves files from disk and copies them to magnetic tape for later recovery. Before an emergency, you must serialize and format the appropriate medium; it is impossible to do so after a system crash.

If the system directory has been destroyed, it may still be possible to save files by using the SADUTIL FIND command to locate files and select those you wish to save. To use the SAVE command when the system directory is intact, proceed as follows:

1. ENTER FUNCTION: SAVE

2. SADUTIL now instructs:

   ```
   READY SERIAL DEVICE FOR WRITE
   ```

   In response to this prompt, mount a previously-formatted or previously-used tape to which the disk files are to be copied (using a new tape results in a Serial Device Failure S59 message).

3. SADUTIL now prompts:

   ```
   FILE NAME (OR LDEV #,%SECTOR ADDRESS)?
   ```

   Indicate the file you want copied to tape. The file must reside on a disk defined during the "SADUTIL Configuration Dialog", or with the CONF function. Indicate a file in one of the following two formats:

   a. *filename*[.*groupname*[.*acctname*]]

      The parameters *filename, groupname,* and *acctname* can be replaced by "@" to signify all members of the set; for example, @.@.MPE or @.UTIL.MPE or @.@.@.

      SADUTIL locates the specified file by searching the system file directory. Before copying the file, SADUTIL proceeds to step 4.

   b. *ldev, sectoraddress*

      The parameter *ldev* is the logical device number of the disk. The *sectoraddress* is the file location.

Find the sector address using the PFIL function (*detail=1*) before using SAVE, or from output previously obtained with the STORE;SHOW or RESTORE;SHOW MPE commands. Use this format when you know that some system file directory information is invalid. If used, SADUTIL takes you to step 5 now.

To terminate the SAVE function, press (Return), and SADUTIL will display the ENTER FUNCTION: prompt.

4. If format (a) was used in step 3, this prompt appears:

   DATE? *mm/dd/yy*

   where:

mm	is a two-digit number representing the month; for example, 06 for June.
dd	is a two-day number representing the day of the month.
yy	is a two-digit number representing the last two digits of the year.

   To omit the date specification and copy all files requested in step 3, press (Return).

   This prompt allows you to restrict retrieval to only those files modified since a specified date. SADUTIL now copies the files requested that were modified on or after the date specified.

   The following information is printed for all files copied: file name, group name, account name, logical device number as configured by SADUTIL, and disk sector address of file label.

   If a single file was selected in step 3, SAVE returns to step 3 now. If @.@.@ was selected in step 3, SAVE terminates here and SADUTIL asks you for a new function. SAVE continues to step 5 in other cases.

   If a file, group, or account specified cannot be located, SADUTIL prints a message to this effect before returning to step 3.

5. If format (b) was selected in step 3, SADUTIL prints the following information for the file, as found in the file label at the specified address: file name, group name, and account name, followed by CONTENTS OF LABEL. Immediately after this information, SADUTIL prints the prompt:

   DO YOU WISH TO RETRIEVE THIS FILE (Y/N)?

   Respond Y or YES to copy the file, or N or NO to avoid copying the file. In either case, SADUTIL performs the requested operation and returns to step 3.

Notes on SAVE

Data is copied to tape in blocks of 128 words. The first block contains the file label, and the remaining blocks contain data and any user labels. Each file is terminated by an end-of-file (EOF) mark. The tape is terminated by an additional EOF.

If SADUTIL encounters a file that does not fit on one reel of tape, it displays the following message:

filename.group.acct --FILE SPANS MORE THAN ONE REEL
MOUNT NEXT REEL

Mount another reel of tape. SADUTIL continues to save the files on the subsequent reel(s). The file is not limited to two reels. It can span any number of tape reels.

Disk sector 28, decimal words 20 and 12/13, must contain the valid directory size and base address, respectively, for the SAVE to be successful. If a disk error is detected when reading sector 28, the following dialog occurs after step 1 in the above series of steps:

Step	Procedure
1.1	CAN'T READ SECTOR 28 OF SYSTEM DISC. STARTING SECTOR OF DIRECTORY?
	Enter the starting address of the system directory.
1.2	NO. OF SECTORS IN DIRECTORY?
	Enter the size of the system directory, in sectors.

If a magnetic tape operation fails, SADUTIL performs the following operations:

1. SADUTIL prints one of the following messages:

```
UNIT WENT OFF/LINE
NOT READY INTERRUPT
TRANSFER ERROR
CMD REJECT
TAPE RUN AWAY
TIMING ERROR
TAPE PARITY ERROR
SERIAL DEVICE FAILURE ldev
```

2. SADUTIL then prints:

```
THE FILES ON THIS TAPE WILL BE SAVED.
PLACE BAD TAPE BACK ONLINE IF IT IS OFFLINE
HIT CR WHEN UNIT IS BACK ONLINE
MOUNT NEXT REEL
```

3. Now, SADUTIL continues to save the files.

Other error messages dealing with reading the disk (such as bad tracks; improper accesses; bad directory addresses or label entries; or files, groups, or accounts not located) may appear. These messages should be self-explanatory. Errors resulting from sectors outside the directory, absence of required account entries, bad tracks, or invalid label comparisons can be investigated by using the EDIT function.

Examples

To retrieve the file SPL.PUB.SYS, use SAVE in one of the following two methods:

```
ENTER FUNCTION:  SAVE
  READY SERIAL DEVICE FOR WRITE
  FILE NAME (OR LDEV#,%SECTOR ADDRESS)? SPL.PUB.SYS  ** By file name**
  DATE? (RETURN)
  SPL     .PUB          .SYS        1          %33560
```

or

```
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? 1,%33560 ** Contents of label **
                                                ** by disk address **
    SPL     .PUB          .SYS
  RETRIEVE THIS FILE(Y/N)? N
  FILE NAME (OR LDEV#,%SECTOR ADDRESS)? (RETURN)
ENTER FUNCTION:
```

STOP

Terminates the SADUTIL program.

Syntax

```
STOP
```

SADUTIL Error Messages

After an error message appears, SADUTIL continues operating and prompts you for further input.

Error Message	Action
ACCOUNT NOT IN DIRECTORY	In a PFIL function request, you referenced a file belonging to an account not present in the system file directory. Re-enter the file reference with the proper account name.
{READ } DISC {WRITE} ERR ON {SEEK } LDEV# *ldev* STATUS=*%status %sectoraddr*	SADUTIL did not complete the input/output operation disk. The *ldev* is the logical device number of the volume, *status* is the hardware status word of the device after the operation, and *sectoraddr* identifies the disk sector where the error occurred. If the error occurred on a removable disk, remove the disk and replace it with a new scratch disk, if possible. Then re-run SADUTIL.
FILE NOT IN DIRECTORY	In a PFIL operation, you referenced a file not present in the system file directory. However, the account and group names were valid and present. Re-enter a correct file name.
FUNCTION NOT COMPLETED	The operation was not completed. Another error message should accompany this.
GROUP NOT IN DIRECTORY	In a PFIL request, you referenced a file belonging to a group not present in the system file directory. However, the account name was valid and present. Re-enter the file reference with the proper group name.
ILLEGAL DEVICE SPECIFICATION	You specified an input/output device for a function not requiring such a specification. Re-enter the function keyword correctly.
INVALID	Check syntax and spelling and try again.
INVALID COMMAND	Invalid EDIT command. Check spelling and syntax, and retry.
INVALID DEVICE SPECIFICATION	You entered the function name correctly, but not the input/output device. Check spelling, and try again.
INVALID DISC ADDRESS	With the EDIT function, you attempted to modify a nonexistent address on the disk. Try again, and enter a correct address.

Error Message	Action
LDEV *ldev* NOT DEFINED	If you forgot to configure *ldev* during the configuration dialog, use the **CONF** function to do it now. Or, check your number, or specify a different logical device.
DRT #*nn* UNIT #*nn* NOT READY	The input/output function cannot be performed because the disk device is not ready. FOr example, a removable cartridge may have been removed. Get the disk ready, and try again.
MISSING DEVICE SPECIFICATION FOR *volumename* VOLUME	The system volume table contains *volumename,* but no such volume has been mounted and configured into the system. Configure the device.
SYSTEM DISC NOT CONFIGURED	This message appears during the initial configuration dialog when the system disk (LDEV 1) has not yet been configured. Proceed with the configuration.
VOLUME *volumename* FOR LDEV #*nn* NOT IN VOLUME TABLE. ENTER VOLUME TABLE ENTRY FOR LDEV:	This message appears when a configured disk is not present in the system volume table. The message only appears during those functions that require all volumes to be mounted (such as **PFIL** and **SAVE**). After you respond, SADUTIL continues with the function. SADUTIL usually keeps track of volume names by reading the system volume table (SVT). But if the SVT is corrupt, SADUTIL asks you the volume table entry for each logical device. Remember that volume numbers are not always the same as logical device numbers. Your responses must correspond to the SVT at the time of the system failure.
INVALID SERIAL DEVICE	You are trying to configure a serial disk which has not been labeled as serial with the VINIT subsystem.
ILLEGAL COPY OPERATION	The specified destination is not a disk, or is the same as the source disk.
ERROR LDEV #*nn* HAS DELETED TRACKS	SADUTIL has scanned the defective track table and found that logical device number *nn* has one or more deleted tracks.
WARNING REMAINING SECTORS OF FROM DISC WILL NOT BE COPIED	The source disk is larger than the destination disk, so excess data cannot be copied.

12

RECOVER2/RECOVER5

RECOVER2/RECOVER5 is used in conjunction with SADUTIL to recover disk files after a serious system crash in which the directory is damaged. SADUTIL, which runs without MPE, stores selected files onto tape without using the directory. Once you have performed a RELOAD to restore the operating system and user files from a SYSDUMP tape, RECOVER2/RECOVER5 can be used to restore the newer files saved by SADUTIL.

Operation

To recover files from a halted system, follow these steps:

1. Use the SAVE function of the SADUTIL utility program to put the necessary files on tape.
2. Reload the operating system and user files from your last full SYSDUMP tape. Refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).
3. If the previous accounting structure cannot be recreated during the RELOAD, create the accounts from the keyboard now. The *MPE V System Operation and Resource Reference Manual* (32033-90005) also contains instructions for creating new accounts.
4. Mount the tape or serial disk that was prepared using SADUTIL, which now contains the files to be copied back into the system.
5. If the files are on tape, put the tape drive online. If the files are on serial disk, use the file equation:

```
:FILE RECOVTP; DEV=SDISC
```

6. Log on and run RECOVER5 as follows:

```
:HELLO MANAGER.SYS
:RUN RECOVER5.PUB.SYS
```

7. RECOVER5 identifies itself and asks you to enter the file sets to recover. Terminate your list with a (Return) in response to the > prompt.

   You may enter file sets in the *file.group.account* format with the following wildcard characters:

   @ Matches from zero to eight characters

   ? Matches any single character

   # Matches any single numeric character (0 to 9)

```
RECOVER5 G.03.00 (C) HEWLETT-PACKARD CO., 1984
ENTER FILESETS TO RECOVER
TERMINATE LIST WITH A NULL LINE
P@.EGAN.MPEM
MORE?> ?SA?#@.TEST.ACCOUNT
MORE?> (RETURN)
```

8. Before recovering any files, RECOVER5 asks (but only once):

```
Do you wish to keep existing copies of files?

Y (or N, as desired).
```

   You must answer this question with one of the choices, or the program stops executing at this point.

9. You are now asked if you wish to restore the ACDs saved during the SADUTIL SAVE process:

```
Do you wish to restore/replace ACDs?
Y (or N as desired)

Do you wish to restore/replace DEVICE ACDs?

Y (or N as desired)
```

10. RECOVER5 begins copying the files, from serial disk or tape, to disk. When the program completes, it lists the names of all of the files that were restored and any error conditions encountered while they were being recovered.

11. You are now asked for other tapes:

```
Is there another recovery tape?
Y (or N, as desired)
```

    If there is another SADUTIL tape to restore, respond Y, mount the tape, and press (Return). Otherwise, RECOVER5 terminates.

12. If RECOVER5 determines that the current file is on multiple reels, it displays the following message:

    filename.group.account - `continues on another tape`
    `Is there a continuation tape for this file?`

    If you answer `YES`, the following message is displayed:

    `Mount a new recovery tape or serial disc`

    Mount the continuation reel(s) for the file. RECOVER5 continues to read the file from the tapes. If you answer `NO` to the question, the file is not recovered and RECOVER5 terminates. If the wrong continuation tape is mounted by mistake, the following message is displayed:

    `The tape mounted doesn't contain the continuation of this file`
    `Is there a continuation tape for this file?`

    If you enter `YES`, the following message is redisplayed:

    `Mount a new recovery tape or serial disc`

    If you enter `NO`, RECOVER5 terminates.

13. When RECOVER5 terminates, you can restore any other files from partial backup tapes by using the `RESTORE` command with the `;KEEP` keyword parameter. Refer to the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

Examples

```
:HELLO MANAGER.SYS
:RUN RECOVER5.PUB.SYS
RECOVER5 G.03.00 (C) HEWLETT-PACKARD CO., 1984
ENTER FILESETS TO RECOVER
TERMINATE LIST WITH A NULL LINE
P@.EGAN.MPEM
MORE?> ?SA?#@.TEST.ACCOUNT
MORE?> (RETURN)
WISH TO KEEP EXISTING COPIES OF FILES? (Y/N) Y
     .
     .
     .
IS THERE ANOTHER RECOVERY TAPE? (Y/N) N

END OF PROGRAM
```

RECOVER2/RECOVER5 Error Messages

The name of the current disk file being recovered is attached to the front of each of the messages which follow. Each RECOVER5 message is followed by a file system error message. For example:

```
WTFILE.DAZO.DATABASE - ERROR ON FOPEN OF EXISTING FILE - CODE 91
EXCLUSIVE VIOLATION: FILE ACCESSED EXCLUSIVELY (FSERR 91)
```

You should examine the file system error messages for additional information about why the error occurred. In the current example, someone tried to recover a file that was being exclusively accessed; the file must be closed by the present user before it may be purged. The RECOVER5 error message gave you a description of what occurred, and the file system error message told you why it happened.

Error Message	Action
DUPLICATE FILE (NOT LOADED)	You have tried to keep existing copies of files being recovered and a file with a duplicate name already exists. If you want to recover the file, purge the existing file or specify to RECOVER5 that you do not want to keep existing copies of files.
ERROR ON FOPEN OF EXISTING FILE - CODE *xxx*	You have tried to purge existing copies of files and RECOVER5 could not open the existing file to purge it. Note the file system error number and take appropriate action.
PURGE ERROR - CODE *xxx*	You have attempted to purge existing copies of files. RECOVER5 was able to open the file but failed when attempting to close the file with the purge option. Note the file system error number and take appropriate action.
FCLOSE FAILURE - CODE *xxx*	RECOVER5 encountered an error while attempting to close a file being saved with the FCLOSE SAVE option. Note the file system error number and take the required action.
LDEV DISC-ADDRESS	This is not an error; it indicates that the file was successfully recovered onto the system. RECOVER5 outputs the logical device number and absolute disk address of the file label after the file name, to indicate successful recovery of the file.
ATTACHIO ERROR: IOCB = *xxxxx*	An I/O error was detected when attempting to read or write the file label of the disk file being recovered. The *xxxxx* is the ATTACHIO error status. Notify your support engineer.
DISC WRITE FAILURE - CODE *xxx*	An error was reported by the file system when writing data to the disk file. Note the file system error number and take appropriate action.
OUT OF DISC SPACE OR TOO MANY RECS ON TP	RECOVER5 discovered more data on the tape than was indicated in the file label. This indicates an internal problem with either RECOVER5 or SADUTIL. Notify your support engineer.
FOPEN FAILED - CODE *xxx*	The FOPEN used to create the disk file to recover failed. Note the file system error number and take appropriate action.
FGETINFO FAILED	The FGETINFO call to obtain file information failed. This indicates a problem with RECOVER5 or MPE. Notify your support engineer.

The error messages which follow do not have file names attached to them. A ** after the message indicates that a file system error message follows the RECOVER5 message.

Error Message	Action
`MOUNT NEXT RECOVERY TAPE`	RECOVER5 has finished with the current serial device medium. Mount a new one, and continue.
`TAPE READ ERROR - CODE` *xxx*	A `READ` error was detected when reading the first block of a file from a serial device. Note the file system error number and contact your support engineer.
`This version will work only on MPE-V`	RECOVER5 runs only on MPE version G.00.00 or later. RECOVER2 runs on earlier versions.
`FAILURE TO OPEN MAG. TAPE FILE **`	The `FOPEN` of the serial device file `RECOVTP` encountered an error. Note the file system error number and take appropriate action.
`CONTROL-Y DETECTED: ABORT OR CONTINUE?`	You have entered Ctrl Y, and now have the option to continue or abort. Enter `C` to continue or `A` to abort.
`IS THERE ANOTHER RECOVERY TAPE (Y/N)?`	RECOVER5 is finished with the current serial device medium, and wants to know if there are more files to recover. If there is another tape or serial disk with data, mount it and continue processing.
`CONTROL-Y WILL BE ACKNOWLEDGED AT THE END OF THIS FILE`	You have entered Ctrl Y, and RECOVER5 is indicating that it will be acknowledged after the current file is recovered.

The following messages are displayed if errors are detected while the file sets are being parsed. A caret (^) is printed under the offending character. Note the error message, and re-enter all file sets.

```
ERROR BUILDING PATTERN IN PATTERNMATCH-NAME

ERROR BUILDING PATTERN IN PATTERNMATCH-GROUP

ERROR BUILDING PATTERN IN PATTERNMATCH-ACCOUNT

EXCESSIVE FILESETS IGNORED (10 ACCEPTED)

PART TOO LONG

ZERO LENGTH PART FOUND

FOUND WILDCARD AND EITHER $ or *

FIRST CHARACTER OF PART MAY NOT BE NUMERIC

LOCKWORD MAY ONLY FOLLOW THE FILE PART

TOO MANY PERIODS WERE FOUND, THE MAXIMUM IS TWO

A * WAS FOUND, BUT IT WASN'T THE FIRST CHAR

BOTH $ AND * MAY NOT BE SPECIFIED

A $ WAS FOUND, BUT IT WASN'T THE FIRST CHARACTER

AN ILLEGAL CHARACTER WAS FOUND

NEEDED PARAMETERS TO THE PROC WERE MISSING
```

13

DISKED2/DISKED5

The utility DISKED5 (DISKED5.PUB.SYS) runs on MPE V/E systems; DISKED2 (DISKED2.PUB.SYS) is the MPE V/P and MPE IV version. The dialog and operation of both are identical. Throughout this chapter, DISKED5 is used in all examples and explanations.

DISKED5 allows you to modify or display the contents of a disk.

Anyone can use the FILE command to operate in file editor mode. This allows you to view and modify any word, except the file label, in your own files. Only users with system manager (SM) capability can modify file labels. A system manager can operate both in disk editor mode, viewing and modifying any word on disk, and in file editor mode.

Caution

The normal safeguards of MPE are bypassed in privileged mode. When attempting to modify privileged data on disk, it is possible to destroy file integrity, or the MPE operating system itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged data, but this service is not included in the standard service contract. Hewlett-Packard will not modify the MPE operating system to accommodate problems arising from your use of DISKED5.

Operation

Follow these steps to use DISKED5.

1. To run DISKED5, type:

```
:RUN DISKED5.PUB.SYS
```

2. DISKED5 identifies itself and prompts you for an input:

```
DISKED5 G.03.00 (C) HEWLETT-PACKARD CO., 1983
TYPE 'HELP' FOR INFO
>
```

3. Respond to the > prompt with a DISKED5 command. Precede octal numbers with the % character; for example, %34.
4. To terminate DISKED5, enter E or EXIT in response to the > prompt.

Commands

DISKED5 has ten commands, each of which is described in detail in the following pages.

BASE	Sets the base sector number used with the MODIFY and DUMP commands.
DEBUG	Calls the MPE Debug facility; PM capability required.
DISC	Specifies the logical device of the file to be modified; SM capability required.
DUMP	Displays selected disk sectors on a list device.
EXIT	Terminates DISKED5.
FILE	Activates the file editor mode; for system managers and regular users.
HELP	Prints a summary of DISKED5 commands.
LIST	Specifies the output device for the DUMP command.
MODIFY	Changes the contents of specified words on disk.
WIDTH	Selects wide or narrow format for output listings.

BASE

Sets the base sector number used with the DUMP and MODIFY commands.

Syntax

BASE *absector*

Parameters

absector A decimal or octal (prefixed with %) number specifying the absolute base sector to which the *relsector* parameter is relative.

Operation

The BASE command, abbreviated B, allows you to dump new sectors of the disk by changing the base sector number from 0 to the octal number you specify. This way, the dump command DUMP 0 will list the contents of the first sector relative to the new base address.

Examples

The example below shows a partial listing (in narrow format) of two different sectors dumped using the identical DUMP command, DUMP 0. The first sector uses a base address of 000. In the second, the base sector address is changed to octal 10 (decimal 8) to the contents of the eighth logical sector. Notice the differences in the dump headings:

```
>DUMP 0

LOGICAL SECTOR 0          *** FILE LABEL ***
SECTOR %00000222632         LDEV = %000002
000:  042111  051513  042504  032440  050125  041040  020040  020040
010:  052125  051116  042522  020040  046501  051103  044501  020040
 .
 .
 .

>BASE %10
>DUMP 0

LOGICAL SECTOR 8
SECTOR %00000222642         LDEV = %000002
000:  020040  020040  020040  020040  020040  020040  020040  020040
010:  020040  020040  020040  020040  020040  020040  020040  020040
 .
 .
 .
```

DEBUG

Calls the MPE Debug subsystem.

Syntax

```
DEBUG
```

Operation

The DEBUG command invokes the MPE Debug facility, described in the *MPE Debug/Stack Dump Reference Manual* (30000-90012). Do not abbreviate DEBUG. You must have privileged mode (PM) capability to use DEBUG.

Type R to exit DEBUG and resume DISKED5.

DISC

Specifies the logical device number of the disk to be modified.

Syntax

DISC *ldev*

Parameters

ldev The logical device number of the disk to be modified. The default is 1.

Operation

The DISC command, abbreviated D, is used to designate which disk you wish to examine and/or edit. To use it, you must be assigned system manager (SM) capability.

You usually specify the logical device number in decimal form. If you choose to enter an octal number, it must be preceded by a percent sign (%).

DUMP

Displays selected disk sectors on your standard list device.

Syntax

```
DUMP { relsector [ ,numsectors ] } [ ,A ]
     { ALL                      }
```

Parameters

relsector	The sector number, relative to the *absector* specified in the BASE command, to use as the starting sector for the dump. The default is zero.
numsectors	The number of sectors to dump. The default is one.
ALL	Specifies that the entire disk is to be dumped.
A	Specifies that the dump is to be in ASCII format.

Operation

The DUMP command dumps the selected sectors onto your standard list device. You may also send the output to the line printer with the LIST command. The starting sector is determined by adding *relsector* to the base sector specified in the *absector* parameter of the BASE command.

Press Ctrl Y to abort the output of the DUMP command.

DUMP may be abbreviated D.

If you do not have system manager (SM) capability, you are prompted for the file name you want to edit.

EXIT

Terminates DISKED5.

Syntax

```
EXIT
```

Operation

The EXIT command may be abbreviated to E.

FILE

Activates the file editor mode.

Syntax

FILE *filename*

Parameters

filename The name of the file to edit. You may use the file name only, or you may fully qualify the file reference by specifying the group and account names.

Operation

Anyone may use the FILE command to view and/or modify his or her own files. Only users assigned system manager (SM) capability can modify a file's label (which is always logical sector 0).

You may abbreviate the FILE command to F.

Examples

In the following example, one sector of the file A.OSE is dumped to the standard list device. In this case, you need not know the sector address of the file, since the file is specified by name:

```
>DISC 1
>FILE A.OSE
>DUMP 0

LOGICAL SECTOR 0          *** FILE LABEL ***
SECTOR %00001207073         LDEV = %000003
000:  040440  020040  020040  020040  041501  051107  047117  047111  A
010:  047523  042440  020040  020040  046101  051122  054440  020040  OSE
020:  020040  020040  020040  020040  020202  004040  000001  124461
030:  124525  124525  000000  000000  010017  000000  000000  001777  .U.U....
040:  000554  000050  011000  022701  000105  177410  000174  000407  .1.(..%.
050:  000200  000200  000000  000001  002405  007073  000000  000000.... ....
   .
   .
   .
150:  000000  000000  000000  000000  004471  025410  124461  000000.... ....
160:  000000  000000  000000  000000  000000  000000  005451  030401.... ....
170:  000000  000000  000000  000000  042111  051503  020234  000001.... ....

>EXIT

END OF PROGRAM
:
```

HELP

Displays a summary of DISKED5 commands.

Syntax

```
HELP
```

Operation

When you first invoke DISKED5, you are prompted to type HELP for assistance. When you do, the following information is displayed:

```
>HELP

DISKED5 allows you to dump and/or modify file contents or
any disk sector (SM capability is required).

 B[ASE] [<ABS SEC #>]
 DEBUG
 DISC <LOG DEV #>
 D[UMP] [ [<REL SEC #>] [, <# OF SEC>] ] OR [<'ALL'>] [, A=ASCII ]
     (AT LEAST ONE PARAMETER MUST BE PRESENT.)
 F[ILE] <FILENAME>
 L[IST] [<DEVICE CLASS>] OR [<LOG DEV #>]
 M[ODIFY] <SEC NUM, REL WORD ADDR [,NUM OF WORDS]>
     (NEW VALUE STARTS WITH : # - DECIMAL, ' - ASCII)
 W[IDTH]
 E[XIT]
```

There are no abbreviations for HELP.

LIST

Specifies the output device for the DUMP command.

Syntax

LIST { *ldev* | *devclass* }

Parameters

ldev	The logical device number where the listings from the DUMP command should appear.
devclass	The device class where the listings from the DUMP command should appear.

Operation

The default list device is $STDLIST. The formal output file designator is DEDILIST. LIST may be abbreviated to L.

To dump the contents of one disk sector and send the output to the line printer instead of $STDLIST, enter:

```
> LIST LP
> DUMP 0
```

To reset the output to $STDLIST, enter the LIST command with no parameters.

MODIFY

Changes the contents of the specified words on disk.

Syntax

MODIFY *sectornum, relwordaddr* [*,numwords*]

Parameters

sectornum A decimal or octal (prefixed with %) number, indicating the logical sector address of the sector to modify. This value is added to the *absector* value specified in the BASE command, if any, to determine the actual sector address to be modified.

relwordaddr A decimal or octal (prefixed with %) number, indicating the address of the first word to modify, relative to the beginning of the sector. The default is %0.

numwords The total number of words to be modified. The default is 1.

Operation

The MODIFY command, which you may abbreviate to M, changes the contents of the specified words on the disk. The address you actually modify is calculated by adding *absector* from the BASE command to the *sectornum* specified in MODIFY. The sector number of each word is displayed, and you are prompted for the contents with a comma (,) in the following form:

SECTOR % *sectornum* LDEV=*%ldev relwordaddr*: *wordcontents*,

In this example, *sectornum* is the absolute sector address, *ldev* is the logical device number, *relwordaddr* is the relative sector address of the word to be modified, and *wordcontents* is the present contents of the specified word.

You may respond in one of two ways:

- Enter a new value for the word, or
- Enter an asterisk (*) or press Return to retain the old value.

Successful changes are confirmed by the message WRITTEN.

Inputs are assumed to be octal unless prefixed with a "#" (such as #4082), which indicates decimal. Input can be ASCII if specified with an apostrophe ('). ASCII input can be directed to either the left byte only, or to both bytes of the word. For example, 'f puts an "f" in the left byte and 'wb puts "w" in the left byte and "b" in the right byte.

To abort MODIFY, type "/" or Ctrl Y.

WIDTH

Selects wide or narrow format for output listings.

Syntax

```
WIDTH
```

Operation

The WIDTH command, which you may abbreviate to W, generates the prompt

```
NARROW FORMAT?
```

You must respond with an uppercase Y or N. Entering Y produces the ASCII character translation of the dump below the listing. If your output device is limited to 80 columns, you must use the narrow format to get a full display.

The default is the wide format, which produces the ASCII translation of the disk sector alongside of the octal listing of the sector. If the octal information in the sector is not a printable ASCII character, dots are printed. To select the wide format, you may either press Return or respond to the question NARROW FORMAT? by entering Y.

Examples

The examples on the next four pages illustrate both the narrow format and the wide format. You may enter the WIDTH command after invoking the DISKED5 utility, or at any time you want to change from the default (wide format) to the narrow format.

These examples use the FILE command to identify the file you wish to examine and/or modify.

This example produces the output in narrow format. To select this format, enter the WIDTH command and answer Y to the question NARROW FORMAT?. The ASCII translation of the octal listing appears directly below the sector dump.

```
> WIDTH
NARROW FORMAT?
Y
>FILE DISKED5.PUB.TURNER
>DUMP 0

LOGICAL SECTOR 0          *** FILE LABEL ***
SECTOR %00000222632         LDEV = %000002
000:  042111  051513  042504  032440  050125  041040  020040  020040
010:  052125  051116  042522  020040  046501  051103  044501  020040
020:  020040  020040  020040  020040  020202  004040  000001  125035
030:  125035  125035  002040  000000  010017  000000  000000  001474
040:  001424  000050  111573  022137  000005  177640  000600  001416
050:  000025  000025  000000  001474  002401  022632  003000  144707
060:  002402  066070  003000  156573  002402  075114  003000  156722
070:  002402  077256  003000  165117  002402  115123  003001  003512
100:  002402  120203  002000  153004  003001  052036  002402  123202
110:  003001  110505  000000  000000  000000  000000  000000  000000
120:  000000  000000  000000  000000  000000  000000  000000  000000
130:  000000  000000  000000  000000  000000  000000  000000  000000
140:  000000  000000  000000  000000  000000  000000  000000  000000
150:  000000  000000  000000  000000  007071  021001  125035  000000
160:  000000  000000  000000  000000  000000  000000  007436  025402
170:  000000  000000  000000  000000  042111  051503  020000  000001

      .1234567.1234567.1234567.1234567.1234567.1234567.1234567.1234567
000:  DISKED5 PUB     TURNER  MARCIA           .. ......... .........<
100:  ...(.{$_..............&<..%.......18...{..zL......~....0...S...J
200:  ..........T........E.........................................
300:  ........................9".................+.........DISC ...
```

Note

In the example above, the numbers along the left margin are word offsets in octal for the octal display, and byte offsets in octal for the ASCII display.

Using the same narrow format, a new sector of the file DISKED5.PUB.TURNER is displayed. The new output is produced by changing the base sector number to octal 10, but using the same DUMP command, DUMP 0. In this example, address 000 displays the contents formerly listed at address 010. You could have dumped this identical sector by entering DUMP %10 using the old base setting, instead of changing the base, as is done here:

```
>BASE %10
>DUMP 0

LOGICAL SECTOR 8
SECTOR %00000222642        LDEV = %000002
000:  020040  020040  020040  020040  020040  020040  020040  020040
010:  020040  020040  020040  020040  020040  020040  020040  020040
020:  020040  020040  020040  020040  020040  020040  020040  020040
030:  020040  020040  020040  020040  030060  030061  032060  030060
040:  056163  070141  061545  020063  020040  020040  020040  020040
050:  020040  020040  020040  020040  020040  020040  020040  020040
060:  020040  020040  020040  020040  020040  020040  020040  020040
070:  020040  020040  020040  020040  020040  020040  020040  020040
100:  020040  020040  020040  020040  020040  020040  020040  020040
110:  020040  020040  020040  020040  030060  030061  032460  030060
120:  057146  063530  044511  044536  071440  020040  020040  020040
130:  020040  020040  020040  020040  020040  020040  020040  020040
140:  020040  020040  020040  020040  020040  020040  020040  020040
150:  020040  020040  020040  020040  020040  020040  020040  020040
160:  020040  020040  020040  020040  020040  020040  020040  020040
170:  020040  020040  020040  020040  030060  030061  033060  030060

      .1234567.1234567.1234567.1234567.1234567.1234567.1234567.1234567
000:                                                          00014000
100:
200:                          00015000
300:                                                          00016000
```

In the following example, the DUMP command is used with the A parameter to produce output in ASCII. The dump begins at sector 0 and dumps a single sector:

```
>FILE JLDMM.WEI.MPEM
>DISC 1
>DUMP 0,1,A

      .1234567.1234567.1234567.1234567.1234567.1234567.1234567.1234567
000:  JLDMM   WEI     MPEM    LIZ                    A               $
100:  .q.(.)................$...V...................................
200:  ..........................................................
300:  .......................23..Y................&.........DISC  ..

>EXIT

END OF PROGRAM
:
```

The following example uses the wide format to display the contents in logical sector 8 of the file DISKED5.PUB.TURNER:

```
> WIDTH
NARROW FORMAT?
N
>FILE DISKED5.PUB.TURNER
>BASE %10
>DUMP 0

LOGICAL SECTOR 8
SECTOR %00000222642       LDEV = %000002
000:  020040  020040  020040  020040  020040  020040  020040  020040
010:  020040  020040  020040  020040  020040  020040  020040  020040
020:  020040  020040  020040  020040  020040  020040  020040  020040
030:  020040  020040  020040  020040  030060  030061  032060  030060
040:  056163  070141  061545  020063  020040  020040  020040  020040
050:  020040  020040  020040  020040  020040  020040  020040  020040
060:  020040  020040  020040  020040  020040  020040  020040  020040
070:  020040  020040  020040  020040  020040  020040  020040  020040
100:  020040  020040  020040  020040  020040  020040  020040  020040
110:  020040  020040  020040  020040  030060  030061  032460  030060
120:  057146  063530  044511  044536  071440  020040  020040  020040
130:  020040  020040  020040  020040  020040  020040  020040  020040
140:  020040  020040  020040  020040  020040  020040  020040  020040
150:  020040  020040  020040  020040  020040  020040  020040  020040
160:  020040  020040  020040  020040  020040  020040  020040  020040
170:  020040  020040  020040  020040  030060  030061  033060  030060
```

There are two ways to specify the printer as the output device for your dump. In the first example, the LIST command is used with your printer's logical device number, #6 in this case:

```
>LIST 6
>DUMP 0
>LIST
>EXIT
```

In the next example, the LIST command is used with your printer's device class name, LP in this case:

```
>DISC 1
>LIST LP
>DUMP 0
>EXIT
```

You may enter DEBUG (which invokes the MPE Debug facility) to isolate a problem. To resume DISKED5 directly from Debug, type R or RESUME at DEBUG's ? prompt, as shown below:

```
>DEBUG

*DEBUG* PRIV.0.1000
    .
    .
    .

?RESUME
```

Once you have found the problem, you can then dump the sector you need to change:

```
>DUMP %300

SECTOR %00000000300        LDEV = %000001
000:  155555  133333  066666  155555  133333  066666  155555  133333
010:  066666  155555  133333  066666  155555  133333  066666  155555
020:  133333  066666  155555  133333  066666  155555  133333  066666
030:  155555  133333  066666  155555  133333  066666  155555  133333
040:  066666  155555  133333  066666  155555  133333  066666  155555
    .
    .
    .
```

Next, the contents of the first two words beginning at the relative octal address 20 are changed. When you enter the MODIFY command, the current contents of the specified word are displayed, followed by a comma and the cursor. Enter the new value at the cursor (next to the current displayed value). DISKED5 confirms the change by displaying WRITTEN below the word.

```
>MODIFY %300,%20,2

SECTOR %00000000300    LDEV = %000001

020:  %133333,1
021:  %066666,2
WRITTEN
```

To verify the changes in the dump, enter:

```
>DUMP %300

SECTOR %00000000300        LDEV = %000001

000:  155555  133333  066666  155555  133333  066666  155555  133333
010:  066666  155555  133333  066666  155555  133333  066666  155555
020:  000001  000002  155555  133333  066666  155555  133333  066666
030:  155555  133333  066666  155555  133333  066666  155555  133333
   .
   .
   .
```

DISKED2/DISKED5 Error Messages

You may receive any of the following messages when you are using DISKED5.

Error Message	Action
****FWRITE ERR ON LIST** *fchecknum*	DISKED5 output is incomplete. The *fchecknum* shows the error code returned by the **FCHECK** intrinsic. Refer to the *MPE V Intrinsics Reference Manual* (32033-90007) for details.
****INVALID DISC ADDRESS****	The **BASE** address is invalid, or relative sector address plus absolute sector address in **DUMP** is less than zero, or sector address plus absolute sector address in **MODIFY** is less than zero, or disk address in any of the three commands is too high for the disk being accessed.
****IRRECOVERABLE DISC ERROR=%0000***nn*	Error during **DUMP** or **MODIFY**. **%0000***nn* specifies the error: **%000004** Invalid function. **%000014** Transmission error. CRC or track-specific error. **%000034** Transmission error. **%000044** SIO not ready. **%000054** Unit failure. All errors other than track-specific. After the disk error number is displayed, DISKED5 returns to prompt (>).
****WRITE ACCESS IS REQUIRED**	You do not have the proper capability to access the file. See your system manager.
****SYS. MGR CAPABILITY IS REQUIRED**	You need system manager (SM) capability to perform this action.
****SYS. MGR CAPABILITY REQUIRED TO MODIFY FILE LABEL**	To modify a file, you must have legal access to it.
****LOCAL DISC FILES ONLY**	Only files on a local machine may be accessed by DISKED5. Files accessed via communications subsystems cannot be accessed by this program.
****HARDWARE END OF FILE ENCOUNTERED**	A hardware error was encountered in reading the input file.

Error Message	Action
**Privileged mode REQUIRED TO ENTER DEBUG	Privileged mode capability is required to enter DEBUG from DISKED5.
DEVICE NOT DISC	A command referenced a device that is not a disk.
DISC IS OFF-LINE	A command referenced a disk that is not online.
UNABLE TO OPEN LIST DEVICE	DISKED5 could not access the referenced device.
WRONG NUMBER OF PARAMETERS	An invalid number of parameters is specified for this command.
FIRST PARAMETER INVALID	The parameter in the position indicated is invalid.
SECOND PARAMETER INVALID	The parameter in the position indicated is invalid.
THIRD PARAMETER INVALID	The parameter in the position indicated is invalid.

14

PATCH

The purpose of PATCH is to give you access to the object code of a program file and to let you display or modify it without recompiling. You can use PATCH to make simple changes to program instructions or to the variables in the global stack area. No special MPE capabilities are required to run PATCH.

Caution

PATCH can bypass normal MPE safeguards and modify the contents of privileged program files. It is possible, therefore, to corrupt a file or the entire MPE operating system. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged software, but this service is not included in the standard service contract. Hewlett-Packard will not modify the MPE operating system to accommodate problems arising from your use of PATCH on privileged files.

The system manager must prevent unauthorized use of PATCH, perhaps by placing a lockword on the program file.

Operation

In order to use PATCH, you must know the memory location where your program symbols are stored, the beginning locations of each program unit, and the offsets from these locations of each line of code. This information is obtained in several ways, depending on the source language of your program.

A FORTRAN program should be compiled with the `MAP`, `LABEL`, and `LOCATION` parameters in the `$CONTROL` command. The `MAP` parameter generates a symbol map that lists all of the symbolic names in your program and the location in the data stack where the data is stored for each symbol. The `LABEL` parameter provides a label map which shows the offset of each labeled statement. The `LOCATION` parameter provides the offset of every statement as part of the source listing.

All programs to be patched should be prepared with the `PMAP` parameter. The floowing material tells you how to prepare your program using the `PMAP` parameter in the `PREP` command.

Using PMAP

Follow these steps to use the PMAP parameter.

1. To use the PMAP parameter with the PREP command, enter the following:

   :PREP *filename,progfilename*;PMAP

   The PREP command and its parameters are described in the *MPE V Commands Reference Manual* (32033-90006). The user subprogram library (USL) file name is $OLDPASS. The use of USL files for compiler output is described in the *MPE Segmenter Reference Manual* (30000-90011). The program file name is $NEWPASS.

   The use of PMAP in connection with the PATCH program is described below. A more general description of PMAP can be found in the *MPE Segmenter Reference Manual* (30000-90011), and in the *MPE Debug/Stackdump Reference Manual* (30000-90012).

2. The PMAP parameter produces information in the following format:

```
PROGRAM FILE progfilename


SEG'              0
   NAME            STT  CODE ENTRY SEG
   OB'               1     0     0
   FOPEN             3                 ?
   PRINTFILEINFO     4                 ?


   FREEDSEG         25                 ?
   TERMINATE'       26                 ?
   QUIT'IT           2  2557       2557
   FCHECK           27                 ?
   FERRMSG          30                 ?
   TERMINATE        31                 ?
   SEGMENT LENGTH       3204

PRIMARY DB          45    INITIAL STACK    2260    CAPABILITY         600
SECONDARY DB       770    INITIAL DL          0    TOTAL CODE        3204
TOTAL DB          1035    MAXIMUM DATA        ?    TOTAL RECORDS       27
ELAPSED TIME   00:00:01.700                      PROCESSOR TIME   00:00.436

END OF PREPARE
:
```

The information has the following meaning:

SEG'	Segment name.
0	Logical segment number.
NAME	Program unit entry point name or external procedure name.
STT	Assigned entry number in segment transfer table (STT).
CODE	Beginning location of procedure code in segment.
ENTRY	Location of entry point in this segment.
SEG	Logical segment number; if ?, the procedure is external to the segment and external to the program file.
PRIMARY DB	The number of words, in octal, used for global stack variables and indirect references to array variables.
INITIAL STACK	The number of words (octal) in the stack when the program first runs. This initial size can increase up to the value specified in the MAXDATA parameter of the PREP command.
CAPABILITY	Capability of the program file, that is, IA, BA, PM, and so on. Refer to word 2 of UCAP in the job information table (JIT) in the *MPE V System Tables Reference Manual* (32033-90010) for the translation of bit position to capability.
SECONDARY DB	The number of words, in octal, used for array elements.
INITIAL DL	The number of words, in octal, that will be allocated to the data limit (DL) area of the stack, as specified in the DL parameter of the PREP command.
TOTAL CODE	Total number of words, in octal, for all of the segments in the program file.
TOTAL DB	The sum of the primary and secondary DB areas.
MAXIMUM DATA	The number of words, in octal, specified in the MAXDATA parameter of the PREP command. A ? in this field means that the USL file was prepared with no maximum data specified. The value defaults to the configured system maximum.

Running PATCH

Follow these steps to run PATCH.

1. To run PATCH, type:

   ```
   :RUN PATCH.PUB.SYS
   ```

2. PATCH identifies itself, then prompts you with ? for the name of the file to list or modify:

   ```
   PROGRAM PATCH G.03.00 (C) HEWLETT-PACKARD CO., 1980
     ENTITY =?
   ```

3. Enter *filename.groupname.accountname,* in uppercase letters, of the file whose code you wish to display or change. An invalid file name terminates PATCH with the following error message:

   ```
   *** ERROR *** UNABLE TO OPEN FILE
   ```

4. When your file name is accepted, you are prompted for a command:

   ```
   ?
   ```

   Respond to the prompt with one of the PATCH commands in uppercase letters only.

Note

PATCH does not verify that the file you specified is a program file. A fatal error results, however, when you attempt to use a command on a nonprogram file.

After each command entry, PATCH prompts you for another command. To terminate PATCH, enter an invalid command or press Return in response to the ? prompt.

D

Displays the contents of a code segment.

Syntax

D, *segnum, address* [,*numlocations*]

Parameters

segnum The logical segment number to be displayed, found at the top of the PMAP output.

address The code segment relative address you wish to display is calculated as follows: add the offset of the instruction within the procedure (an octal value found in your compiler listing) to the procedure start address found in the CODE column of your PMAP output.

numlocations The number of words, in octal, to be displayed.

M

Modifies the contents of a code segment.

Syntax

M, *segnum, address* [,*numlocations*]

Parameters

segnum	The logical segment number to be modified, found at the top of the PMAP output.
address	The address of the code segment you wish to modify, calculated by adding the offset of the instruction within the procedure (an octal value found in your compiler listing) to the procedure start address found in the CODE column of your PMAP output.
numlocations	The number of words, in octal, to be modified.

Operation

When you enter the M command, the contents of the current instruction are displayed followed by a comma. To leave the value unchanged, you must re-enter the contents. To set the instruction to a NOP (%000000), press Return.

Examples

The following example shows you how to display and modify the contents of one instruction in the file PIOCONF.PUB.FUDD:

```
ENTITY = ? PIOCONF.PUB.FUDD

?D,0,20,1
  031042
?M,0,20,1
  031042, RETURN
?D,0,20,1
  000000
?E
```

DG

Displays the global area of the initial stack.

Syntax

DG, *reloffset* [,*numwords*]

Parameters

reloffset The DB-relative offset of the word to display, found in your compiler listing.

numwords The number of words, in octal, that you wish to display. The default is 1.

Operation

At the end of the compiler listing is a list of global identifiers for the program file and their addresses. For example:

```
IDENTIFIER          CLASS           TYPE            ADDRESS
BUFF                ARRAY           LOGICAL         DB+001
CLEANRINTABLE       PROCEDURE
FCLOSE              PROCEDURE                       DB+002
FNUM                SIMP. VAR.      LOGICAL
FOPEN               PROCEDURE       INTEGER
FREADDIR            PROCEDURE                       DB+000
GETSIR              PROCEDURE       LOGICAL
NAME                ARRAY           BYTE            DB+003
PRINT               PROCEDURE
QUIT                PROCEDURE       INTEGER
RINSIR              SIMP. VAR.      LOGICAL
SETCRITICAL         PROCEDURE
TERMINATE           PROCEDURE
WRITEDSEG           PROCEDURE


PRIMARY DB STORAGE=%004;            SECONDARY DB STORAGE=%00211
NO. ERRORS=0000;                    NO. WARNINGS=000
PROCESSOR TIME=0:00:03;             ELAPSED TIME=0:00:21
```

Examples

In the following example, the integer RINSIR is initialized to 38 (octal 46). For arrays, the value displayed is the pointer to the actual contents of the array:

```
FILE? PINITRIN.PUB.FUDD

?DG,0,4
   000010
   000015
   000000
   000046
?E
```

MG

Modifies the global area of the initial stack.

Syntax

MG, *reloffset* [,*numwords*]

Parameters

reloffset	The DB-relative offset of the word to modify, found in your compiler listing.
numwords	The number of words, in octal, that you wish to modify. The default is one.

Operation

The MG command displays the contents of the current stack word followed by a comma. To leave the value unchanged, you must re-enter the contents. To fill the word with zeros, press Return.

Examples

In this example, all commands are used to illustrate a typical application of PATCH:

```
PROGRAM PATCH G.03.00 (C) HEWLETT-PACKARD CO., 1980
  ENTITY =?$OLDPASS

?DG, 0,5    ** Displays the values at the first five locations. **
  000112
  000052
  000064
  000264
  000464

?MG, 0,5
  000112,
  000052,    ** Each time you press Return, the value is set to zero. **
  000064,
  000264,
  000464,
```

```
?DG ,0,5      ** Displays the result of the previous '?MG' command. **
  000000
  000000
  000000
  000000
  000000

?MG,0,5
  000000,112      ** Contents changed to 112. **
  000000,52
  000000,64
  000000,264
  000000,464

?DG,,1             ** Displays the first changed value. **
  000112

END OF PROGRAM
:
```

PATCH Error Messages

When PATCH encounters a fatal error, it terminates.

15

SLPATCH

SLPATCH allows you to display or modify the contents of a segmented library (SL) file. This utility requires no special user capabilities, but does run within standard MPE security restrictions. Before you use SLPATCH to alter code, refer to the *MPE Segmenter Reference Manual* (30000-90011).

Caution

SLPATCH can bypass normal MPE safeguards and modify the contents of privileged SLs. It is possible, therefore, to corrupt a file or the entire MPE operating system. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged software, but this service is not included in the standard service contract. Hewlett-Packard will not modify the MPE operating system to accommodate problems arising from your use of SLPATCH on privileged files.

The system manager must prevent unauthorized use of SLPATCH, perhaps by placing a lockword on the program file.

Running SLPATCH

To use SLPATCH, you need the segment names and code displacements of your SLs. Request a PMAP of your SL by using the PREP parameter of the -ADDSL Segmenter command. The PMAP output shows you what your SL looks like and verifies your segment names.

The segment displacement is found by adding the instruction offset (found in your compiler listing) to the starting address of the procedure, found on your PMAP listing. All input to SLPATCH must be in uppercase letters.

1. To run SLPATCH, type:

   ```
   :RUN SLPATCH.PUB.SYS
   ```

2. SLPATCH identifies itself, then prompts you with ? for the name of the SL file you wish to modify or display. Enter the SL file name, one to eight alphanumeric characters long, in the form *filename*[.*groupname*[*acctname*]]; for example:

   ```
   SLPATCH G.03.00 (C) HEWLETT-PACKARD CO., 1980
   SL FILE? SL.BOOKS.DOC
   ```

3. When your SL designator is accepted, you are prompted with ? for a segment name and a command. SLPATCH prompts you for another command after each response. To terminate SLPATCH, type EXIT in response to the ? prompt.

D

Displays the contents of an SL segment.

Syntax

[*segname*,] D, *segdisplace* [,*numwords*]

Parameters

segname The name of the segment you want to display, obtained from an SLCREF (SL cross-reference) listing. This parameter must be specified the first time the segment is accessed, but may be omitted in subsequent commands.

segdisplace The segment displacement, calculated as described in "Running SLPATCH".

numwords The number of words of the SL to be displayed. The default is 1.

M

Modifies the contents of an SL segment.

Syntax

[*segname*,] M, *segdisplace* [,*numwords*]

Parameters

segname	The name of the segment you want to modify, obtained from an SLCREF (SL cross-reference) listing. This parameter must be specified the first time the segment is accessed, but may be omitted in subsequent commands.
segdisplace	The segment displacement, calculated as described in "Running SLPATCH".
numwords	The number of words of the SL to be modified. The default is 1.

Operation

The contents of each word to be modified are displayed, followed by a comma. To retain the old value, type "*" and press Return. To enter a new value, type the number (in octal) and press Return. If you simply press Return, you are prompted to enter "*" or a number.

Example

```
:RUN SLPATCH.PUB.SYS
SLPATCH G.03.00 (C) HEWLETT-PACKARD CO., 1980

SL FILE? SL.BOOKS   ** A group-level SL is available **
                          ** to everyone in the group **
? SDMCOMM,D,0,4
025001
051404
041605
021040

? D,0,3         ** This command is also within the **
                        ** SDMCOMM segment, so the **
025001                  ** segment name is not repeated. **
051404
041605

? M,0,1         ** Change the contents of the first **
                        ** word of the segment. **
025001,025002

? EXIT

END OF PROGRAM
:
```

SLPATCH Error Messages

Fatal errors encountered when using SLPATCH may result in one of the following error messages:

```
xxIO ERROR***

***END OF FILE***

FILE ERROR=xx
```

In these messages, *xx* is an error code returned by the FCHECK intrinsic, as described in the *MPE V Intrinsics Reference Manual* (32033-90007).

After the message is printed, SLPATCH terminates and MPE prints the following message:

```
ABORT :SLPATCH.PUB.SYS.%n.%nnnn
PROGRAM ERROR #18 :PROCESS QUIT .PARAM = 1
 END OF PROGRAM
```

Non-fatal errors may result in one of the following messages:

```
ILLEGAL COMMAND

INVALID SL FILE

SEGMENT NOT SPECIFIED

ILLEGAL RANGE

ILLEGAL SEGMENT NAME

ILLEGAL NUMBER
```

Press Return to get the ? prompt, then enter the correct information.

16

VINIT

Run the VINIT subsystem to format, initialize, and label private volumes, serial disks, and foreign disks. VINIT requires system supervisor (OP) or system manager (SM) capability.

Operation

Running VINIT

You may initiate VINIT from a session or job, but not in BREAK or from a program. Pressing [BREAK] suspends execution.

To run VINIT online, type the following:

```
VINIT listdevice
```

or

```
VINIT
```

where *listdevice* is the name of the device to receive output generated by the VINIT subsystem. The *listdevice* parameter must backreference a FILE command that names the device. The default *listdevice* is $STDLIST.

Note

In JOB mode, the VINIT subsystem can be invoked only as follows:

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

The prompt in the VINIT subsystem is ">". Once accessed, VINIT displays the prompt, and awaits a command from you. Brief definitions of these commands and their parameters are listed below.

Note

It is advisable to perform VINIT only after a full backup. If a system failure occurs during a VINIT, you must do a RELOAD to recover.

VINIT commands and parameters are discussed in detail under the heading "Private Volumes" in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

Commands

>COND

```
>COND ldev[ ;RECOVER]
```

Condenses free disk space and recovers lost disk space for private volumes.

Note You cannot recover disk space on a private volume containing open files.

>COPY

```
>COPY fromldn,toldn [ ;GEN=[ genindex] ]
                    [ ;VER[ IFY] ]
                    [ ;BUF=size  ]
```

Copies the contents of one volume to another.

>DSTAT

```
>DSTAT [ ldev ]
       [ ALL  ]
       [ @    ]
```

Displays the status of disk drives.

>DTRACK

```
>DTRACK ldev
```

Processes suspect tracks in the defective track table and defective sectors in the defective sectors table. As you use DTRACK, you can either recover (by ignoring), delete, or reassign the defective tracks.

>EXIT

```
>EXIT
```

Exits the VINIT subsystem.

>FOREIGN

```
>FOREIGN ldev
```

Makes a disk volume foreign by filling the label area of a volume with zeros.

>FORMAT

```
>FORMAT ldev[ ;IBM[ :volid] ]
```

Allows online formatting of disk packs.

>HELP

```
>HELP
```

Prints a listing of the VINIT commands, including syntax and descriptions, at $STDLIST.

>INIT

```
>INIT vname,ldev[ ,vsname,group,acct ]
                [ ;GEN=genindex      ]
```

Completes the conditioning of a private volume previously formatted by the SLEUTHSM program or the >FORMAT command.

>PDEFN

```
>PDEFN [*.group.account vsname.group.account]
```

Lists the specified volume set. The asterisk (*) specifies the home volume set. The default is the home volume set.

>PDTRACK

```
>PDTRACK ldev
```

Prints the contents from the defective tracks table or the defective sectors table at $STDLIST.

>PFSPACE

```
>PFSPACE { ldev[ ;ADDR ] }
         { ALL          }
```

Prints the disk free-space map at $STDLIST.

>PLABEL

```
>PLABEL ldev
```

Prints the disk label at $STDLIST.

>SCRATCH

>SCRATCH *ldev* [;RESET]

Writes a scratch-type label, making the volume available for use.

>SERIAL

>SERIAL *ldev*

Writes a serial disk-type label.

>VERIFY

>VERIFY *ldev*[;DTT]

Verifies the readability of data.

Command Parameters

ldev	The logical device number of the disk drive on which the volume is mounted.
fromldn	The logical device number of the disk from which data is copied. Unless the device is a system domain disk, you must first remove it from normal system use by issuing a DOWN command for the device.
toldn	The logical device number of the disk volume to which data is copied. The device must be in the DOWN state. You must first remove the device from normal system use by issuing a DOWN command for the device.
genindex	A value from 0 to 32767 inclusive, indicating the generation index of the disk volume. If not specified, the *genindex* of *toldn* will be one greater than the *genindex* of *fromldn*. This parameter is valid only for private volumes.
VER[IFY]	Checks the readability of data on the destination drive. If an error is detected, VER[IFY] directs the VINIT subsystem to attempt to recover the track. If it is unrecoverable, the track is reassigned and VINIT attempts to copy data once again. The VERIFY keyword may be abbreviated as VER.
size	The internal buffer size, which may be a value from 1 to 32 inclusive.
IBM[:*volid*]	Indicates that the disk is a single-sided IBM-formatted diskette. If *volid* is specified, it denotes an IBM volume identifier other than the default *volid* of IBMIRD.
vname	The name of the disk volume that is a member of the volume set.
vsname.group.acct	The name of the volume set.
ADDR	Displays the disk free space entries for the specified logical device.
ALL or A	Displays the disk free-space map in table format for all systems and private volumes that are physically mounted.
RESET	Resets a volume from scratch to non-scratch.
DTT	Directs VINIT to verify only suspect sectors and/or tracks as recorded in the defective sectors and/or tracks tables. This option is not available for serial or foreign disks or cartridge tapes.

Example

```
FILE LIST;DEV=LP
VINIT *LIST
```

In this example, the file LIST is directed to the lineprinter. VINIT is run, and its output is sent to the backreferenced file, LIST.

A

Capabilities and Access Modes

Capability Sets

Capability sets are assigned by the system manager. They determine the MPE functions and command set that each user may access. Your capability set is the result of your user attributes, file attributes, and program attributes.

User Attributes

User attributes designate your level of access to MPE system resources.

SM	System manager
AM	Account manager
AL	Account librarian
GL	Group librarian
DI	Diagnostician
OP	System supervisor
NA	Network administrator
NM	Node manager
PS	Programmatic creation of sessions

File Attributes

File attributes determine your file/device interface capabilities.

SF	Save permanent files
ND	Access of non-shareable I/O devices
CV	Create volume sets
UV	Use nonsystem domain volume sets (which includes private volumes).
CS	Use communication subsystems
LG	User logging

Program Attributes

Program attributes determine your data structure resources (intrinsics, system tables, and instructions).

PH	Process-handling
DS	Extra data segments
MR	Multiple RINs
PM	Privileged mode
IA	Interactive access
BA	Local batch access

Access Mode Abbreviations

R	READ. Allows you to read files.
L	LOCK. Lets you prevent concurrent access to a file by yourself and another user. Specifically, it means that you can use the FLOCK and FUNLOCK intrinsics, and the exclusive access option of the FOPEN intrinsic, as described in the *MPE V Intrinsics Reference Manual* (32033-90007).
A	APPEND. Allows you to add data and disk extents to files, but prevents you from altering or deleting information already written. This access mode implicitly allows you to use the LOCK (L) access mode.
W	WRITE. Allows you to add, modify, or delete information on files. You can remove entire files from the system with the PURGE command while in this mode. The WRITE (W) access mode also implicitly gives you LOCK (L) and APPEND (A) access modes.
S	SAVE. Allows you to make existing files within your group into permanent files with the SAVE command or the FCLOSE intrinsic. You may rename files with the RENAME command. You may also create new permanent files with the BUILD command.
X	EXECUTE. Permits you to run program files with the RUN command or the CREATE intrinsic.

User-Type Abbreviations

ANY	Any user. Any user defined in the system, including all categories defined below.
AL	Account librarian. A user who can manage all files in the account, regardless of their groups.
GL	Group librarian. A user who can manage certain files within the home group.
CR	Creator. The user who created a specific file.
GU	Group user. Any user whose home group is this group, including all GL users applicable to this group.
AC	Account member. Any user within an account, including the AL, GU, and CR users of this account.

Index

A

Aborting commands
  Control Y, 10-3
Accessing MPE commands from SPOOK, 4-3
Accessing object code, 14-1
Accessing SL files, 15-1
Accessing spool files, 4-31
Account attributes, 3-5
ACDs, 3-10
Administrative utilities, 1-2
ALTER command (SPOOK), 4-5
APPEND command (SPOOK), 4-7
ASOCTABL/ASOCTBL5, 9-1
  error messages, 9-4
Attributes
  account, 3-5
  account/file/group/user, 3-1
  files, 3-8
  group, 3-6
  user, 3-7
  user, file and program, 1-1
Auditing user activity, 8-6

B

Backing up the SDF, 10-1, 10-4
BASE command (DISKED5), 13-3
Batch listing
  in DPAN5, 10-6

C

Capabilities
  for using ASOCTBL5, 9-1
  VINIT, 16-1
Capability set
  user, 1-1
Changing output priority
  spool file, 4-5
Characteristics of spool files, 4-28
Character strings
  locating, 4-13
CLID command (SADUTIL), 11-8
Coldloading DUS, 11-3
Coldload Medium
  preparation, 11-2
COLDSTART option
  in DPAN5, 10-6
Command parameters
  VINIT, 16-6
Commands
  help (SPOOK), 4-15
  LISTDIR5 , 3-2
  SADUTIL, 11-1
  SPOOK, 4-2
  VINIT, 16-1
Commands in DISKED5, 13-2
Command syntax of SPOOK
  describing, 4-32
COND command (VINIT), 10-1, 16-2
CONF command (SADUTIL), 11-9
Console
  dismount request, 3-13
  mount request, 3-12
Contiguous free space, 5-1
Control data
  displaying, 4-21
Control-Y
  use with HELP command, 3-15
Control Y
  aborting current command, 10-3
COOLSTART option
  in DPAN5, 10-6
COPY command (SADUTIL), 11-10
COPY command (SPOOK), 4-9
COPY command (VINIT), 16-2
Copying multiple files, 4-10
Creating a device class/user association table, 9-1

D

Damaged directory
  recovering disk files, 12-1
Data formats
  memory error logs, 6-2
D command (PATCH), 14-5
D command (SLPATCH), 15-2
DEBUG
  capability, 4-11
DEBUG command (DISKED5), 13-4
DEBUG command (SPOOK), 4-11
Deleting child process, 4-18
Deleting spool files, 4-24
Device class table

format, 4-2
Device class/user association table
creating, 9-1
Device file ID, 4-3
Device types
SADUTIL, 11-7
DG command (PATCH), 14-7
Diagnostic utility system, 11-3
DISC command (DISKED5), 13-5
DISKED2/DISKED5, 13-1
error messages, 13-18
DISKED5
commands, 13-2
Disk files
recovering from damaged directory, 12-1
Disk operations
after system failure, 11-1
Disks
displaying/modifying contents, 13-1
Disks, serial and foreign
format/initialize/label, 16-1
Disk usage
free space, 5-1
DISMOUNT command (LISTDIR5), 3-13
Displaying
control data, 4-21
Displaying disk contents, 13-1
Displaying object code, 14-1
Displaying SL files, 15-1
DPAN4/DPAN5, 10-1
error messages, 10-11
DSTAT command (VINIT), 16-2
DTRACK command (VINIT), 16-3
DUMP command (DISKED5), 13-6
Dumping memory data to a device, 10-1
DUS, 11-3

E

EDIT command (SADUTIL), 11-11
Emergency disk operations
system failure, 11-1
Error logs, memory
printing, 6-1
Error messages
ASOCTABL/ASOCTBL5, 9-4
DISKED2/DISKED5, 13-18
DPAN4/DPAN5, 10-11
FREE2/FREE5, 5-2
LISTDIR2/LISTDIR5, 3-16
LISTEQ2/LISTEQ5, 2-2
LISTLOG2/LISTLOG5, 8-8
MEMLOGAN, 6-8
MEMTIMER, 7-2
PATCH, 14-10
RECOVER2/RECOVER5, 12-4
SADUTIL, 11-28
SLPATCH, 15-4
SPOOK4/SPOOK5, 4-33
Error recovery utilities, 1-2
Executing program files, 4-26
EXIT command (DISKED5), 13-7
EXIT command (LISTDIR5), 3-14
EXIT command (SPOOK), 4-12
EXIT command (VINIT), 16-3

F

File attributes, 3-8
listing on private volumes, 3-4
FILE command (DISKED5), 13-1, 13-8
File equations
listing , 2-1
File modification utilities, 1-2
Files
access control definition, 3-10
ASSOCIATE, 9-3
emergency storage, 11-1
listed in the system directory, 3-4
listing, 4-19
locating character strings, 4-13
security provisions, 3-10
Files, disk
recovering after system crash, 12-1
Files, program
accessing object code, 14-1
Files, SL
accessing contents, 15-1
Files, system log
analyzing, 8-1
FIND command (SADUTIL), 11-13
FIND command (SPOOK), 4-13
FOREIGN command (VINIT), 16-3
Foreign disks
format/initialize/label, 16-1
Formal file designator, 3-2
LISTEQ, 2-1
FORMAT command (VINIT), 16-3
Formatted tables
listing, 10-6
Formatting private volumes/serial/foreign disks, 16-1
FREE2/FREE5, 5-1
error messages, 5-2
Free space
on disk, 5-1

G

General system utilities, 1-1
Generic names, 3-3
Group attributes, 3-6

H

HELP
  prompt, 4-15
HELP command (DISKED5), 13-9
HELP command (LISTDIR5), 3-15
HELP command (SADUTIL), 11-15
HELP command (SPOOK), 4-15
HELP command (VINIT), 16-3

I

INFO string
  use in DPAN5, 10-7
INIT command (VINIT), 16-4
Initializing private volumes/serial/foreign disks, 16-1
INPUT command (SPOOK), 4-16
Interactive listing
  in DPAN5, 10-6

K

KILL command (SPOOK), 4-18

L

Labeled tapes
  copying multiple files, 4-10
Labeling private volumes/serial/foreign disks, 16-1
Line width
  adjusting, 4-21
LISTACCT command (LISTDIR5), 3-5
LIST command (DISKED5), 13-10
LIST command (SPOOK), 4-19
LISTDIR2/LISTDIR5, 3-1
  error messages, 3-16
LISTDIR5
  security restrictions, 3-2
LISTDIR5 commands, 3-2
  stopping/suspending, 3-2
LISTEQ2/LISTEQ5, 2-1
  error messages, 2-2
LISTEQ5
  formal file designator, 2-1
LISTF command (LISTDIR5), 3-8
LISTGROUP command (LISTDIR5), 3-6
Listing attributes, 3-1
Listing data segments, 10-6
Listing file equations, 2-1
Listing main memory, 10-1
Listing security provisions, 3-1
Listing spool files, 4-1
Listing temporary files, 2-1
LISTLOG2/LISTLOG5, 8-1
  Error messages, 8-8
LISTSEC command (LISTDIR5), 3-10
LISTUSER command (LISTDIR5), 3-7
Loading SADUTIL, 11-3
Log files, 8-1
  printing histories, 8-2
Logging intervals
  setting, 7-1

M

M command (PATCH), 14-6
M command (SLPATCH), 15-3
MEMLOGAN, 6-1
  error messages, 6-8
Memory dump
  analysis options, 10-6
  batch listing, 10-6
  dumping data, 10-1
  interactive listing, 10-6
  listing storage and capability, 10-1
  Series 37, 10-1
  Series 39/40/42/44/48/52/58, 10-4
  Series 64/68/70, 10-5
Memory error logging, 6-1
  data formats, 6-2
  setting intervals, 7-1
MEMTIMER, 7-1
  error messages, 7-2
MG command (PATCH), 14-9
MODE command (SPOOK), 4-21
MODIFY command (DISKED5), 13-11
Modifying disk contents, 13-1
Modifying disk files, 1-2
Modifying object code, 14-1
Modifying SL files, 15-1
MOUNT command (LISTDIR5), 3-12
MPE commands
  accessing from SPOOK, 4-3
MPE Debug facility, 4-11

N

NAME option (LISTLOG5), 8-6
Names
  generic, 3-3
  using wildcard characters, 3-3

O

Object code
  accessing/displaying/modifying, 14-1
Options for analyzing memory dump, 10-6
Options for DPAN5, 10-6
OUTM command (SADUTIL), 11-16
OUTPUT command (SPOOK), 4-22
Output device directory
  format, 4-2
Output files

deleting, 4-24
storing, 4-22
Output priority
changing for spool file, 4-5

P

Parameters
VINIT command, 16-6
PATCH, 14-1
error messages, 14-10
PDEFN command (VINIT), 16-4
PDSK command (SADUTIL), 11-17
PDTRACK command (VINIT), 16-4
PDTT command (SADUTIL), 11-19
PFIL command (SADUTIL), 11-20
PFSPACE command (VINIT), 16-4
PLABEL command (VINIT), 16-4
Preparing a Coldload Medium
SADUTIL, 11-2
Printing memory error logs, 6-1
Private volume
dismount request, 3-13
mount request, 3-12
Private volumes
format/initialize/label, 16-1
free space, 5-1
listing file attributes, 3-4
Privileged mode, 4-11
Program files
executing, 4-26
Prompt
HELP and SPOOK, 4-15
Prompt (
), 2-1
PURGE command (SPOOK), 4-24
PVOL command (SADUTIL), 11-22

Q

QUIT command (SPOOK), 4-25

R

Range list
syntax, 4-4
RECOVER2/RECOVER5, 12-1
error messages, 12-4
RECOVER5, 11-1
Recovering disk files, 12-1
Recovering files after system failure, 11-2
RELOAD option
in DPAN5, 10-6
RESTORE
use after RECOVER5, 12-3
Restoring output spool files, 4-16
RUN command
in DPAN5, 10-6
RUN command (MPE)
deleting processes, 4-18
RUN command (SPOOK), 4-26
Running SADUTIL, 11-3
Running SPOOK, 4-3
Running VINIT
online/batch, 16-1

S

SADUTIL, 11-1
device types, 11-7
error messages, 11-28
SADUTIL commands, 11-1
SAVE command (SADUTIL), 11-23
SCRATCH command (VINIT), 16-5
SDF
dumping memory, 10-1
Security precautions (SLPATCH), 15-1
Security provisions
files, 3-10
listing, 3-1
Security Restrictions
LISTDIR5, 3-2
Segmented library files
accessing/displaying/modifying, 15-1
SERIAL command (VINIT), 16-5
Serial disks
format/initialize/label, 16-1
Series 37
memory dump, 10-1
Series 39/40/42/44/48/52/58
memory dump, 10-4
Series 64/68/70
memory dump, 10-5
Setting logging intervals, 7-1
SHOW command (SPOOK), 4-28
SL files
accessing/displaying/modifying, 15-1
SLPATCH, 15-1
error messages, 15-4
Software dump facility, 10-1
backing up, 10-1, 10-4
SPOOK4/SPOOK5
error messages, 4-33
SPOOK commands, 4-2
stopping/suspending, 4-3
SPOOK/SPOOK4/SPOOK5, 4-1
SPOOK tape format, 4-2
Spool file, output
changing parameters, 4-5
Spool files
accessing, 4-31
appending, 4-7
configuring in DPAN5, 10-1

copying, 4-9
deleting from system, 4-24
device file ID, 4-3
listing, 4-19
listing characteristics, 4-28
listing/manipulating/transferring, 4-1
Spool files, output
restoring, 4-16
storing, 4-22
$STDLIST, 2-1
STOP command (SADUTIL), 11-27
Stopping LISTDIR5 commands, 3-2
Stopping SPOOK commands, 4-3
Storing files to tape
emergency procedures, 11-1
Storing spool files, 4-22
STREAM command
in DPAN5, 10-6
Suspending LISTDIR5 commands, 3-2
Suspending SPOOK commands, 4-3
Suspending SPOOK operation, 4-12
Syntax
listed by XPLAIN, 4-32
Syntax for a range list, 4-4
System directory
file listing, 3-4
System failure
emergency disk operations, 11-1
listing memory, 10-1
System log files
analyzing, 8-1
System utilities, 1-1

T

Table
device class/user association, 9-1
Tape format
SPOOK, 4-2
Temporary files
default, 2-1
listing, 2-1
Terminating LISTDIR5, 3-14
Terminating PATCH, 14-4, 15-1
Terminating SPOOK, 4-25
Terminating SPOOK operation, 4-12
TEXT command (SPOOK), 4-31
Total free space, 5-1
Transferring spool files, 4-1

U

Unlabeled tapes
copying multiple files, 4-10
UPDATE option
in DPAN5, 10-6
User activity
auditing individuals, 8-6
User association/device class table, 9-1
User attributes, 3-7
User capability set, 1-1
USE VOLUME capability, 3-4
Using ASOCTBL5
capabilities, 9-1
Using DPAN5
configuring spool files, 10-1
Using RESTORE after RECOVER5, 12-3
Using SPOOK
multiple versions, 4-2
Utilities
definition, 1-1

V

VERIFY command (VINIT), 16-5
Versions of SPOOK, 4-2
VINIT, 16-1
command parameters, 16-6
running online or in batch mode, 16-1
user capabilities, 16-1
VINIT commands, 16-1
Virtual storage
listing, 10-1

W

WARMSTART option
in DPAN5, 10-6
Warnings (SPOOK), 4-33
WIDTH command (DISKED5), 13-12
Wildcard characters, 3-3, 3-8

X

XPLAIN command (SPOOK), 4-32

Part No. 32033-90008
Printed in U.S.A. 1989
E1089

HEWLETT PACKARD