

HP 3000 Series II Computer System



MPE System Utilities

Reference Manual



5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA, 95050

NOTICE

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

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

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 program language without the prior written consent of Hewlett-Packard Company.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

Pages	Effective Date	Pages	Effective Date
Title	Mar 1977	7-1 to 7-2.	Mar 1977
ii to x	Mar 1977	8-1 to 8-10.	Mar 1977
1-1 to 1-35.	Mar 1977	9-1 to 9-3.	Mar 1977
2-1 to 2-2.	Mar 1977	10-1 to 10-3.	Mar 1977
3-1 to 3-3	Mar 1977	11-1 to 11-34	Mar 1977
4-1 to 4-3.	Mar 1977	12-1 to 12-5.	Mar 1977
5-1	Mar 1977	13-1 to 13-2.	Mar 1977
6-1 to 6-3.	Mar 1977		

PRINTING HISTORY

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition change.

First Edition. Jun 1976

Second Edition. Mar 1977

This manual is one of the set of manuals that document the Multiprogramming Executive Operating System (MPE-II) for the HP 3000 Series II Computer System. It describes the following MPE utility programs:



PROGRAM	FUNCTION
SADUTIL	Performs various disc operations, including copying disc files to tape for later retrieval.
RECOVER2.PUB.SYS	Re-creates, on disc, files copied to tape by SADUTIL.
DPAN2.PUB.SYS	Prints analytical, formatted listing of main-memory dump.
LISTLOG2.PUB.SYS	Prints listing of MPE Log File contents.
LISTEQ2.PUB.SYS	Lists :FILE commands and session/job temporary files used during current session/job.
DISKED2.PUB.SYS	Modifies and lists contents of disc files.
FREE2.PUB.SYS	Prints histograms of available space on each disc in system, and total number of entries in Free-Space Table for discs.
LISTDIR2.PUB.SYS	Lists attributes of accounts, groups, users, and files.
PATCH.PUB.SYS	Displays or modifies contents of program file.
SLPATCH.PUB.SYS	Displays or modifies contents of segmented library (SL) files.
SPOOK.PUB.SYS	Interrogates and operates on MPE spooled device-files (spoolfiles).
MEMLOGAN.PUB.SYS	Displays error history recorded by Memory Error Logging System.
MEMTIMER.PUB.SYS	Changes time interval between successive updates of memory log-file.

All but one of these programs reside in the Public Group (PUB) of the System Account (SYS), and run under the control of MPE. The remaining program, SADUTIL, is a stand-alone program cold-loaded into the system from magnetic tape.

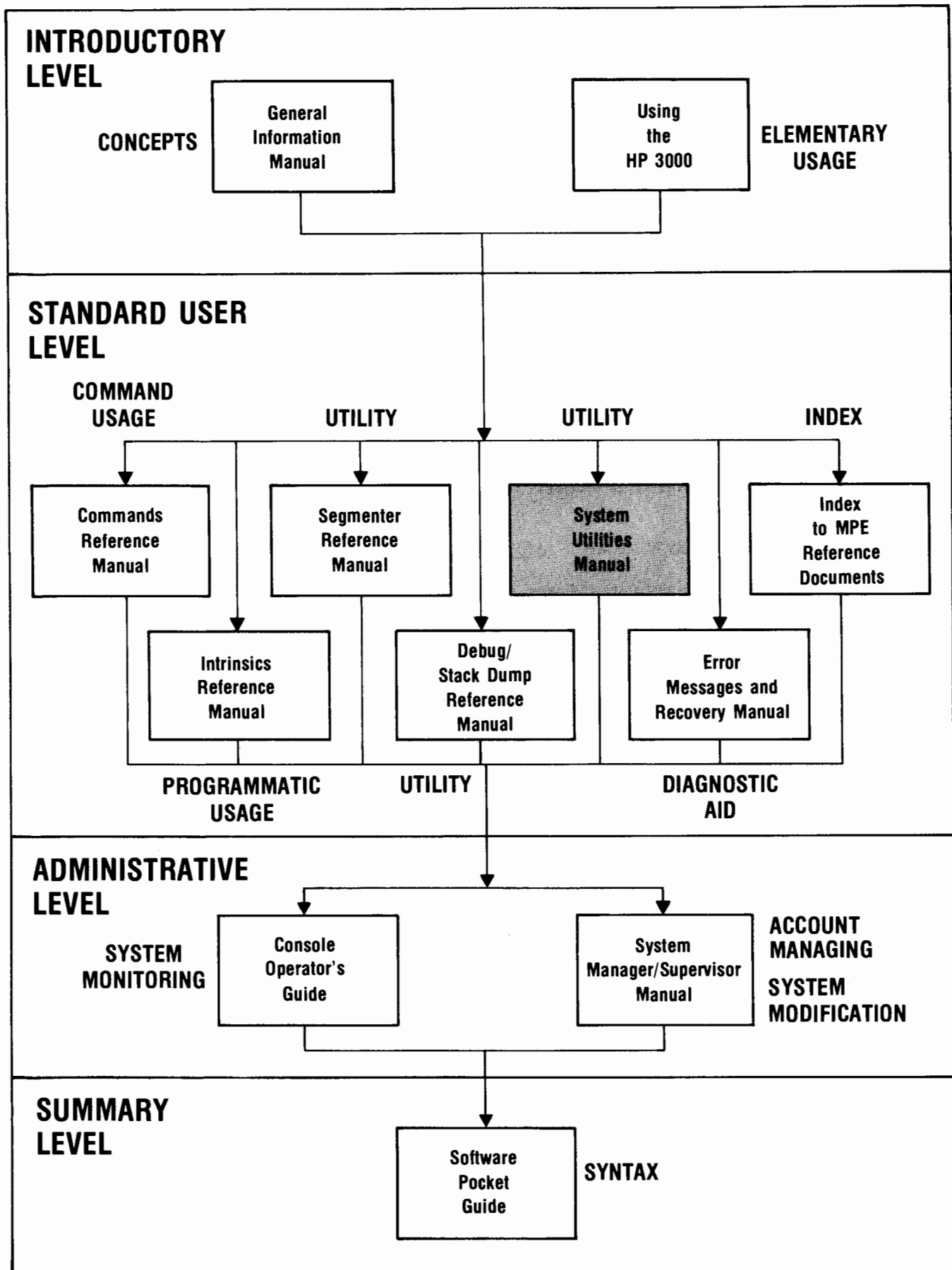
The Manual Plan on page vii shows the relationship of this manual (shaded block) to others in the MPE manual set. Further information about the Manual Plan appears in *Index to MPE Reference Documents* (part no. 30000-90045). The following manuals contain information you may require while using the programs described in this manual:

MANUAL TITLE	PART NO.
Stand-alone Diagnostic Utility Program II (SDUPII) Reference Manual	03000-90125
MPE Commands Reference Manual	30000-90009
MPE Intrinsic Reference Manual	30000-90010
MPE Debug/Stackdump Reference Manual	30000-90012
Console Operator's Guide	30000-90013
System Manager/System Supervisor Manual	30000-90014
System Error Messages and Recovery Manual	30000-90015

NOTE

The normal checks and limitations that apply to the standard users in MPE are bypassed in privileged mode. It is possible for a privileged mode program to destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of privileged mode code. This service, which is not provided under the standard service contract, is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.

MANUAL PLAN



CONVENTIONS USED IN THIS MANUAL

NOTATION

DESCRIPTION

[]

An element inside brackets is *optional*. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements.

Example: $\begin{bmatrix} A \\ B \end{bmatrix}$ user may select A or B or neither

{ }

When several elements are stacked within braces the user **must** select one of these elements.

Example: $\begin{Bmatrix} A \\ B \\ C \end{Bmatrix}$ user must select A or B or C.

italics

Lowercase italics denote a parameter which must be replaced by a user-supplied variable.

Example: CALL *name*
name one to 15 alphanumeric characters.

underlining

Dialogue: Where it is necessary to distinguish user input from computer output, the input is underlined.

Example: NEW NAME? ALPHA1

superscript C

Control characters are indicated by a superscript C

Example: Y^C

return

return in italics indicates a carriage return

linefeed

linefeed in italics indicates a linefeed

...

A horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.

CONTENTS

Section I	Page	Section VIII	Page
SADUTIL	1-1	LISTDIR2	8-1
SADUTIL Operations.	1-1	Commands.	8-1
Print Functions.	1-1	Security Restrictions	8-9
Specialized Functions.	1-2		
Utility Functions.	1-2	Section IX	Page
Creating a SADUTIL Cold-Load Tape	1-2	PATCH	9-1
Cold-Loading SADUTIL	1-6		
SADUTIL/Operator Dialogue	1-7	Section X	Page
Functions	1-9	SLPATCH	10-1
COND.	1-11		
CONF.	1-13	Section XI	Page
EDIT	1-15	SPOOK	11-1
OUTM.	1-17	SPOOK Operations	11-1
PDSK	1-19	Entering Commands.	11-3
PDTT	1-21	ALTER.	11-5
PFIL.	1-23	DEBUG.	11-7
PFRE	1-25	EXIT	11-9
PVOL	1-27	FIND	11-11
SAVE	1-29	INPUT	11-13
STOP	1-33	LIST.	11-15
Error Messages	1-34	MODE.	11-17
		OUTPUT.	11-19
Section II	Page	PURGE.	11-21
RECOVER2.	2-1	SHOW.	11-23
		TEXT	11-27
Section III	Page	XPLAIN	11-29
DPAN2	3-1	Messages	11-31
Section IV	Page	Section XII	Page
LISTLOG2.	4-1	MEMLOGAN	12-1
		Memory-Logging Operations	12-1
Section V	Page	Running MEMLOGAN	12-1
LISTEQ2.	5-1	MEMLOGAN Output.	12-2
		Error Messages	12-5
Section VI	Page		
DISKED2	6-1	Section XIII	Page
		MENTIMER.	13-1
Section VII	Page		
FREE2	7-1		

ILLUSTRATIONS

Title	Page	Title	Page
Creating a SADUTIL Cold-Load Tape	1-5	LISTDIR2 Command Examples	8-5
SADUTIL/Operator Dialogue	1-9	PATCH Example.	9-2
LISTLOG2 Dialogue	4-3	SLPATCH Example	10-3
LISTEQ2 Input/Output	5-1	MEMLOGAN Output.	12-4
FREE2 Input/Output.	7-2		

TABLES

Title	Page	Title	Page
Creating a SADUTIL Cold-Load Tape	1-2	LISTDIR2 Commands	8-3
Cold Load from Standard Magnetic Tape	1-6	Capability-Set Abbreviations	8-8
SADUTIL/Operator Dialogue	1-7	Access-Mode Abbreviations	8-8
Using RECOVER2.	2-1	User-Type Abbreviations.	8-9
Obtaining Memory Dump	3-1	Using PATCH.	9-1
CIR Contents After Dump	3-2	PATCH Error Numbers.	9-3
Obtaining DPAN2 Listing	3-2	Using SLPATCH	10-1
Obtaining LISTLOG2 Output	4-1	SPOOK Messages.	11-31
DISKED2 Commands.	6-1	MEMTIMER Error Messages	13-2
DISKED2 Error Messages	6-3		

SADUTIL is a stand-alone utility program that allows you to request various disc operations. When used in conjunction with the RECOVER2 utility program (for re-creating disc files, as described in Section II), SADUTIL can be employed to recover files from an HP 3000 Series II System that has become logically inoperable because of some catastrophic condition. (Such a condition might arise from invalidity of system file directory or cold-load information.) For disc file recovery, see the SAVE function described later in this section.

NOTE

Because SADUTIL is a stand-alone program that runs without operating system control, no special MPE capability is required to run it. This program is, however, typically used by System Managers and Supervisors.



WARNING

When you run this program, you are not operating under control of MPE and you may inadvertently destroy the system. Use this program only in emergencies or after the entire system has been backed-up.

SADUTIL OPERATIONS

When SADUTIL is cold-loaded and initiated following system termination, you can request the various functions through function keywords entered during an interactive dialogue with SADUTIL. These functions are grouped into three categories: print, specialized, and utility functions, summarized below and discussed later in this section.

PRINT FUNCTIONS

Print functions allow various disc tables or the contents of a disc itself to be printed on the line printer or system console. All print functions begin with the letter P.

- *PDSK* prints an octal or ASCII dump of any given area on a specified disc volume.
- *PDTT* prints the defective tracks table of a specified disc volume.
- *PFRE* prints the disc free-space map for a specified disc volume.
- *PVOL* prints information contained in the volume label of a specified disc volume.
- *PFIL* prints descriptions of files contained in the system file directory.

SPECIALIZED FUNCTIONS

Specialized functions perform the following disc operations:

- *COND* condenses the data on a disc volume.
- *EDIT* modifies the contents of a disc volume.
- *SAVE* recovers files from disc, and copies them to magnetic tape.

UTILITY FUNCTIONS

Utility functions aid in controlling the internal operations of SADUTIL:

- *OUTM* sets the output mode for the print functions to console or line-printer output.
- *CONF* re-enters the device configuration dialogue for the configuration of additional devices.
- *STOP* terminates the SADUTIL utility program.

CREATING A SADUTIL COLD-LOAD TAPE

SADUTIL is loaded into the system from a stand-alone magnetic tape. If no tape containing this program exists at your site, you should create one by using the Stand-alone Diagnostic Utility Program II (SDUPII). This program copies SADUTIL to tape from a program file on disc. To run SDUPII, proceed as shown in Table 1-1.

Table 1-1. Creating a SADUTIL Cold-Load Tape

STEP	PROCEDURE
1	Log-on to the system. Then run SDUPII by entering: :RUN SDUPII.PUB.SYS
2	SDUPII identifies itself by printing its title and a prompt that allows you to request instructions on how to use the program: 3000 DIAGNOSTIC UTILITY PROGRAM (SDUP) D417A.00.00 (C) COPYRIGHT HEWLETT-PACKARD COMPANY 1976 DO YOU WANT INSTRUCTIONS? ANSWER 'YES' OR 'NO' Enter YES to request SDUPII to display the instructions at your terminal (as shown in Figure 1-1) or NO to suppress this display.
3	SDUPII now prompts you for the type of diagnostic you wish to copy to tape: INPUT DIAGNOSTIC TYPE Respond by entering <u>2</u> , which indicates an input/output stand-alone diagnostic.

4	SDUPII requests you to enter the name of the SADUTIL program file on disc: PROGRAM NAME? Respond by entering: <u>SADUTIL.PUB.SYS</u> NOTE At this point, you can request the SDUPII pre-configuration option by appending the suffix ;C to the program name. This option allows you to configure disc devices into the system in Step 9 of this dialogue rather than in the dialogue that takes place when SADUTIL is running. In Step 9, you enter the configuration information directly into the stack addresses that will be used by SADUTIL. To request this option, enter: <u>SADUTIL.PUB.SYS;C</u> You can only request this option with a Type 0 (moving-head) disc; to configure a fixed-head disc, you must run SADUTIL.
5	SDUPII repeats the PROGRAM NAME? prompt; respond by entering a slash-mark (/) to continue with the program.
6	SDUPII prompts you for the device reference table (DRT) number of the line printer used for listing output: INPUT DRT OF THE LINE PRINTER A CARRIAGE RETURN ASSUMES NO LINE PRINTER Reply with the DRT number of the line printer, or with a carriage-return to indicate that no line printer is used. To indicate an octal value, precede the number with a percent (%) sign. If you indicate no line printer, the program transfers to Step 8.
7	SDUPII prompts you for the type of line printer used: INPUT LINE PRINTER TYPE MODELS-2607A, 2613A, 2617A, AND 2618A ARE TYPE 0 MODELS-2610A, AND 2614A ARE TYPE 1 A CARRIAGE RETURN ASSUMES TYPE = 0 Specify the line printer type by entering a one-digit code—0 or 1— as directed in the prompt. NOTE These are the same type values accepted by SADUTIL in its configuration dialogue.
8	SDUPII prints the following message on your terminal, stating that a request to mount the cold-load tape has been directed to the console operator: MOUNT TAPE ON UNIT TAPE REQUEST HAS BEEN ISSUED OPERATOR MUST NOW REPLY TO REQUEST The operator is requested to mount the tape file named SDUP, and allocate the tape unit to your session.

9	If you did not request the pre-configuration option in Step 4, SDUPII transfers to Step 10. If you did request this option, SDUPII prompts you for pre-configuration specifications to be entered onto its stack. PRECONFIGURATION CHANGES FOR PROGRAM FILE SADUTIL.PUB.SYS;C DB LOCATION TO BE CHANGED? Reply by entering the first of the DB locations to be changed, as noted below. The locations contain the following information: DB+1 contains configuration of Logical Device No. 1. DB+2 contains configuration of Logical Device No. 2. . . DB+16 contains configuration of Logical Device No. 16. Each location noted above contains the following information for each disc device: <table data-bbox="597 842 980 1003"> <thead> <tr> <th>Bits</th><th>Contents</th></tr> </thead> <tbody> <tr> <td>(0:8)</td><td>DRT number.</td></tr> <tr> <td>(8:4)</td><td>Subtype.</td></tr> <tr> <td>(12:4)</td><td>Unit number.</td></tr> </tbody> </table> Indicate octal values by prefixing them with %. As you enter each DB address, SDUPII prints: DB LOCATION yyy = zzzzzz NEW VALUE In this message, yyy is the DB location requested and zzzzzz is the present contents of that location (initially zeros). Respond by entering the octal value for the item to be configured. Again, indicate octal values by the prefix %. After you enter each value, SDUPII returns the DB LOCATION TO BE CHANGED? prompt. Enter another DB location or terminate pre-configuration by entering a carriage-return. (The SDUPII/SADUTIL configuration of logical devices need not correspond with that of the standard system.)	Bits	Contents	(0:8)	DRT number.	(8:4)	Subtype.	(12:4)	Unit number.
Bits	Contents								
(0:8)	DRT number.								
(8:4)	Subtype.								
(12:4)	Unit number.								
10	When SDUPII writes SADUTIL to the tape, it displays the following information on your terminal: • Program number (1) and name (SADUTIL.PUB.SYS). • Logical segment number (which appears on PMAP when requested from the MPE Segmenter). • Physical segment number (assigned by SDUPII to the logical segment while loading). • PB address of segment.								
11	SDUPII prints the following message and terminates: END OF PROGRAM								

An example of SDUPII/user dialogue appears in Figure 1-1.

```

:RUN SDUPIL.PUB.SYS

3000 DIAGNOSTIC UTILITY PROGRAM(SDUP)D417A.01.00
(C) COPYRIGHT HEWLETT-PACKARD COMPANY 1976

DO YOU WANT INSTRUCTIONS?
ANSWER 'YES' OR 'NO'

YES
THE DIAGNOSTIC UTILITY PROGRAM(SDUP30) IS A PROGRAM
THAT CONVERTS DIAGNOSTIC PROGRAM FILES TO A COLD LOADABLE TAPE
THIS TAPE WILL BE USED TO COLD LOAD STAND-ALONE DIAGNOSTICS.
THERE ARE 2 TYPES OF STAND-ALONE DIAGNOSTICS

TYPE 1 IS A CPU DIAGNOSTIC WHICH HAS ONLY ONE SEGMENT AND
USES A SIMPLIFIED LOADER

TYPE 2 IS A STAND-ALONE DIAGNOSTIC WHICH MAY HAVE MORE
THAN ONE SEGMENT AND WILL BE LOADED WITH A RELOCATING LOADER.
THIS TYPE DIAGNOSTIC MAY ALSO HAVE DIALOGUE TO THE SYSTEM CONSOLE.

THE PROGRAM WILL REQUEST DIAGNOSTIC TYPE
REPLY '1' FOR CPU OR '2' FOR STANDALONE
THE PROGRAM WILL REQUEST THE NAME OF THE PROGRAM FILE
REPLY WITH THE NAME OF THE PROGRAM FILE
TYPE ANY CHARACTER TO CONTINUE
return

IF YOU WANT TO USE THE PRECONFIGURATION OPTIONS
TYPE A ';' AFTER THE PROGRAM NAME
A '/' TERMINATES THE PROGRAM INPUT PHASE

INPUT DIAGNOSTIC TYPE
2
PROGRAM NAME?
SADUTIL.PUB.SYS;C
PROGRAM NAME?
/
INPUT DPT OF THE LINE PRINTER
A CARRIAGE RETURN ASSUMES NO LINE PRINTER
14
INPUT LINE PRINTER TYPE
MODELS-2607A,2613A,2617A,AND 2618A ARE TYPE 0
MODELS 2610A,AND 2614A ARE TYPE 1
A CARRIAGE RETURN ASSUMES TYPE = 0
0
MOUNT TAPE ON TAPE UNIT

TAPE REQUEST HAS BEEN ISSUED
OPERATOR MUST NOW REPLY TO REQUEST

PRECONFIGURATION CHANGES FOR
PROGRAM FILE      SADUTIL.PUB.SYS;C

DB LOCATION TO BE CHANGED?
11
DB LOCATION 11= 000000
NEW VALUE =
1002501
DB LOCATION TO BE CHANGED?
return

01 SADUTIL.PUB.SYS;C

LOGICAL SEG  PHYSICAL SEG  PB
000005      000003      010000
000000      000004      023701
000001      000005      030226
000002      000006      034577
000003      000007      043624
000004      000010      047765
000006      000011      050632

END OF PROGRAM

```

Figure 1-1. Creating a SADUTIL Cold-Load Tape

COLD LOADING SADUTIL

Before you can request any SADUTIL functions, you must cold load and initiate SADUTIL as described in table 1-2.

Table 1-2. Cold Load From Standard Magnetic Tape

STEP	PROCEDURE
1	Mount the cold-load tape containing SADUTIL on the tape unit configured as Device Reference Table (DRT) Entry No. 6, and press the 0 switch (bottom left of tape unit panel).
2	Place all other tape units off-line, or set them to any unit number but zero (by pressing the 1, 2, or 3 switches on these units).
3	Press the RESET button and the LOAD button on the DRT No. 6 tape unit. The tape winds forward and stops at the load point; the LOAD light turns ON.
4	Press the ON-LINE button on the DRT No. 6 tape unit.
5	Set the system switch register on the system control panel to %003006.
6	Press the ENABLE and LOAD switches on the system control panel simultaneously.
7	Set the system switch register to %000001. This indicates that the first program on the cold-load tape, SADUTIL, is to be loaded.
8	Press the RUN switch. This loads SADUTIL and re-winds the tape.
9	When the tape is re-wound, press the <i>return</i> key on the console keyboard. In response, SADUTIL begins execution, initiating a dialogue with you by way of the console. This dialogue is described under <i>SADUTIL/Operator Dialogue</i> , below.

SADUTIL/OPERATOR DIALOGUE

The SADUTIL/Operator Dialogue proceeds as shown in table 1-3 below, with SADUTIL output shown in upper-case letters. This dialogue allows you to list all devices pre-configured into SADUTIL, if any, and to add devices to the configuration or remove them from it. (Pre-configuration is accomplished via the Stand-alone Diagnostic Utility Program (SDUPII) for the Series II, as shown in table 1-1.) The last step in the SADUTIL/Operator Dialogue, Step 5, allows you to select particular SADUTIL functions. Such functions, however, can only operate on devices that you can list in Step 3 or 4.6.

Table 1-3. SADUTIL/Operator Dialogue

STEP	DIALOGUE
1	DEVICE CONFIGURATION CHANGES? If you respond YES (or Y), the dialogue continues in sequence. Otherwise, SADUTIL skips to Step 5.
2	PRINTER DRT? To change or configure the line printer, enter the DRT number (in decimal) of the desired line printer and proceed to Step 2.1. Otherwise, enter a carriage <i>return</i> and proceed to Step 3.
2.1	TYPE? If you are changing or configuring an HP Model No. 2607A, 2613A, 2617A, or 2618A printer, enter 0 (a zero). If you are changing or configuring an HP Model No. 2610A or 2614A printer, enter 1.
3	LIST LOGICAL DEVICES? To print a listing of all the logical devices, as well as the DRT entry number, unit number, type, and sub-type of each device, enter YES (or Y). To suppress this listing, enter NO (or N) or a carriage <i>return</i>.
4	DISC CONFIGURATION CHANGES? To prepare for changes to the disc configuration, enter YES (or Y). Otherwise, enter a carriage <i>return</i> and skip to Step 5.
4.1	LOGICAL DEVICE? To specify a device to be added or removed, enter the logical device number (in decimal) of that device. To skip to Step 4.6, enter a carriage <i>return</i>.

Table 1-3. SADUTIL/Operator Dialogue (Continued)

STEP	DIALOGUE
4.2	DRT? To add a device, enter its DRT entry number. To delete a device and return to Step 4.1, enter a zero.
4.3	UNIT? Enter the physical hardware unit number (in decimal) of the device to be added.
4.4	TYPE? Enter the MPE device type (in decimal) for the logical device to be added.
4.5	SUB-TYPE? Enter the MPE device sub-type (in decimal) for the logical device to be added. SADUTIL then transfers to Step 4.1.
4.6	LIST LOGICAL DEVICES? To print a listing of all the logical devices, as well as the DRT entry number, unit number, type, and sub-type of each device, enter <i>YES</i> (or <i>Y</i>). To suppress this listing, enter <i>NO</i> (or <i>N</i>) or a carriage <i>return</i>.
5	ENTER FUNCTION? To execute a particular function, enter the appropriate function name. This step is repeated until the <i>STOP</i> function is entered; then SADUTIL prints the following message on the console and halts: END OF PROGRAM

Figure 1-2 provides an example of the SADUTIL/Operator Dialogue.

```

HP3000/30 DISC UTILITY (A.00.00)
DEVICE CONFIGURATION CHANGES? Y
PRINTER DRT? return
LIST LOGICAL DEVICES? Y
  LDEV  DRT  UNIT  TYPE  SUBTYPE
  ----  ---  ----  ----  -
      1    15    0    0    5
      2    15    0    0    4
DISC CONFIGURATION CHANGES? Y
LOGICAL DEVICE? 6
DRT? 32
UNIT? 0

TYPE? 0
SUBTYPE? 4
LOGICAL DEVICE? return
LIST LOGICAL DEVICES? Y
  LDEV  DRT  UNIT  TYPE  SUBTYPE
  ----  ---  ----  ----  -
      1    15    0    0    5
      2    15    0    0    4
      6    32    0    0    4
ENTER FUNCTION:

```

Figure 1-2. SADUTIL/Operator Dialogue

FUNCTIONS

The SADUTIL functions are described in the following pages. For easy reference, they are presented alphabetically by function name. For each function, the following information is shown:

- **SYNTAX** shows the format of the function request.
- **NOTES** describes in detail the function and related special considerations.
- **EXAMPLES** shows applications of the functions.

You request each function by entering it in response to the *ENTER FUNCTION:* prompt in the SADUTIL/Operator Dialogue.

In the syntax descriptions for the SADUTIL functions, all optional parameters are enclosed in [brackets].

NOTE

As a general syntax rule, the parameter *ldn* appearing in several functions is the logical device number assigned to the disc volume on which these functions are to operate. If this parameter is omitted, the logical device number is assumed to be 1 (the system disc) by default.

COND

Condenses the data on a disc volume.

SYNTAX

COND [*ldn*]

NOTES

This function alters the useable disc space by:

- Reducing the number of free areas on the disc.
- Increasing the average size of the free areas remaining.

The *ldn* parameter specifies the logical device number of the disc volume on which the condense operation is to occur.

When you execute this function, you are asked if there are any factor changes. A factor, with respect to the COND function, means a limitation which will affect the completeness of the condense operation. The less complete a particular condense operation is with respect to the space obtained, the less time it will take to execute. The only factor which can be modified by you is the size factor. This factor is the minimum size (in sectors) of an adjoining free area that will be considered for possible condensing. When the factor of zero (the default) is entered, SADUTIL attempts to condense the entire volume.

EXAMPLES

The following example shows how to condense data on the disc:

```
ENTER FUNCTION: PFRE 1 ← Lists areas on ldn 1 that you may condense.
LDEV= 1, DRT= 15, UNIT= 0, TYPE= 0, SUBTYPE= 5
NO. OF ENTRIES: 5
TOTAL VOLUME CAPACITY: 19200 SECTORS
TOTAL FREE SPACE AVAILABLE: 1436 SECTORS
MAXIMUM CONTIGUOUS AREA: 1074 SECTORS
NO. ADDRESS SIZE
---
1. 13208 1
2. 13384 3
3. 14198 7
4. 14215 351
5. 18126 1074

ENTER FUNCTION: COND 1 } Condenses free areas on ldn 1 which exist
FACTOR CHANGES? Y } in groups of 100 sectors or greater.
SIZE FACTOR? 100 }
ENTER FUNCTION: PFRE 1 ← Lists areas on ldn 1 after they are
LDEV= 1, DRT= 15, UNIT= 0, TYPE= 0, SUBTYPE= 5
NO. OF ENTRIES: 4
TOTAL VOLUME CAPACITY: 19200 SECTORS
TOTAL FREE SPACE AVAILABLE: 1436 SECTORS
MAXIMUM CONTIGUOUS AREA: 1425 SECTORS
NO. ADDRESS SIZE
---
1. 13208 1
2. 13384 3
3. 14198 7
4. 17775 1425
ENTER FUNCTION:
```


CONF

Re-enters the device configuration dialogue for the configuration of additional devices (except line printer).

SYNTAX

CONF [*ldn*]

By entering the CONF function, you will re-enter the following configuration dialogue.

- 1 DEVICE CONFIGURATION CHANGES?
- 2 LOGICAL DEVICE?
- 3 DRT?
- 4 UNIT?
- 5 TYPE?
- 6 SUBTYPE?
- 7 LIST LOGICAL DEVICES?

(Refer also to table 1-3.)

The line printer can only be configured through the initial configuration dialogue and not through the CONF function.

If the *ldn* parameter is specified, only that device will be configured and the dialogue will begin at step 3. If no *ldn* is specified, you are able to configure as many devices as you wish.

EXAMPLES

To configure *ldn* 6 (after the initial SADUTIL configuration dialogue has been completed), follow the steps shown in the dialogue below:

```
ENTER FUNCTION: CONF 6
DRT? 32
UNIT? 0

TYPE? 0
SUBTYPE? 4
LIST LOGICAL DEVICES? Y
  LDEV   DRT   UNIT   TYPE   SUBTYPE
  ----   ---   ----   ----   -
      1    15     0      0      5
      2    15     0      0      4
      6    32     0      0      4
ENTER FUNCTION:
```


EDIT

Modifies the contents of a disc volume.

SYNTAX

EDIT

NOTES

This function prompts you with a greater-than sign (>) when EDIT commands are to be entered. You may then request the type of disc modifications desired by typing one of the following EDIT commands. If you respond to the prompt with a carriage *return*, SADUTIL exits from the EDIT function and prompts you for another function.

> *OUTM* $\left\{ \begin{array}{c} C \\ P \end{array} \right\}$

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

> *PDSK*

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

> *DISC* [*ldn*]

Specifies a logical device on which the volume to be edited exists. (Default: 1.)

> *BASE* [*basesector*]

Specifies a base sector address relative to which all disc address references in the PDSK and MODIFY commands are made (default is 0).

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

Permits you to modify specified words on the disc volume (default for *discaddr* is 0, *wordloc* is 0, and *wordcount* is 1).

SADUTIL responds to the MODIFY request by printing the following:

```
SECTOR %  sectoraddr
          wordloc : oldcontents : =
```

where

sectoraddr is the sector address (in octal). This is the absolute address, and is not relative to *basesector*.

wordloc is the location (in octal) of the word to be modified in the sector.

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

The new contents of the word should be entered as a string of six or less octal characters. If a "/" is entered in place of the octal string, the EDIT function will terminate the current MODIFY operation and prompt you for a new command.

The second MODIFY response above (*wordloc:oldcontents:=*) will be printed for each of the words to be modified.

Where the contents of a disc sector have been modified, SADUTIL will print "WRITTEN" on the console before prompting for the next command.

EDIT

(Continued)

EXAMPLES

To modify 1 word of sector 0 (relative to *baseaddr*) beginning at word 7 ____

```
ENTER FUNCTION: EDIT
>DISC 2
>OUTM 0
>BASE 0
>MODIFY 0,7,1
SECTOR % 0
7: 012451: = return (Contents not changed.)
```

To modify 3 words of sector 2 (relative to *baseaddr*) beginning at word 2 ____

```
>MODIFY 2,2,3
SECTOR % 1
2: 000014: = return } (Contents not changed.)
3: 000004: = /
```

To modify 1 word (third parameter is missing, default is 1) of sector 1 (relative to *baseaddr*) beginning at location 2 ____

```
>MODIFY 1,2
SECTOR % 1
2: 000015: = 14 } (Contents changed.)
WRITTEN
```

To modify 3 words of sector 0 (relative to *baseaddr*) beginning at location 126 ____

```
>MODIFY 0,126,3
SECTOR % 0
176: 000000: = 0 } (Contents not changed.)
177: 000000: = 0
SECTOR % 1
0: 000003: = 2 } (Contents changed.)
WRITTEN
```

(Since there are only 128 words per sector, the third word will be in the next sector.)

NOTE

In the above examples, SADUTIL writes the message *WRITTEN* whenever data is actually written to disc. This only occurs when you enter a numeric value rather than *return* or a slash-mark (/) to the right of at least one MODIFY response, and the new data differs from the old data.

OUTM

Directs the output for the print functions
to the console or the line printer.

SYNTAX

OUTM $\begin{bmatrix} C \\ P \end{bmatrix}$

NOTES

The parameter C directs output to the console, while the parameter P directs output to the line printer. If the parameter is omitted, C is assumed. SADUTIL always begins its execution in C mode.

The P format produces output in lines of 132 characters, suitable for a line printer. The C format prints lines of 72 characters, suitable for a terminal.

EXAMPLES

To set the output mode to the line printer ____

ENTER FUNCTION: OUTM P

To set the output mode to the console ____

ENTER FUNCTION: OUTM

or ____

ENTER FUNCTION: OUTM C

PDSK

Prints an octal/ASCII dump of
any given area contained on the
specified disc volume.

SYNTAX

PDSK [*ldn*]

NOTES

The parameter *ldn* is the logical device number assigned to that volume.

This function repeatedly prompts you with the following message:

ENTER ADDRESS:

You may respond with the following:

$$lsect1 \left[\begin{array}{c} ,numsect \\ :lsect2 \end{array} \right] \left[\begin{array}{c} A [, O] \\ O [, A] \end{array} \right]$$

where

lsect1 is the starting sector number within the logical device.

numsect is the number of sectors to be printed.

lsect2 is the ending sector number within the logical device.

A prints the dump in ASCII.

O prints the dump in octal.

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

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

The printing of the dump may be stopped at any time by setting the high order bit (bit 0) in the system switch register. After detecting this bit setting, the function will ask you to enter another address. A carriage return entered at this point will cause the function to terminate.

EXAMPLES

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

```
ENTER FUNCTION: PDSK 1
ENTER ADDRESS: 0
ENTER ADDRESS:
```

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
```


PDTT

Prints the Defective Track Table
of a specified disc volume.

SYNTAX

PDTT [*ldn*]

NOTES

The Defective Track Table, used by MPE to maintain a record of any defective areas on a disc, resides in the second sector (Sector 1) of each volume. It is written in the following format:

Word	Contents
0	Number of entries (n) in the table (ranges from 0 to 120).
1-n (Bits 0:14) (Bits 14:2)	Track number. 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 discs only).
127	Logical disc pack size (in cylinders for moving-head discs, and tracks for fixed-head discs).

EXAMPLES To print the Defective Track Table of the volume residing on logical device 1____

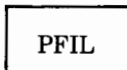
ENTER FUNCTION: PDTT 1

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:

SYNTAX



NOTES

This function repeatedly prompts you with the following message:

ENTER NAME:

You may respond with the following:

$$\left\{ \begin{array}{l} \text{fileset } [, \text{detail}] \\ \text{, detail} \end{array} \right\}$$

where:

$$\text{fileset} = \left[\begin{array}{c} \text{file} \left[\begin{array}{c} \text{. group} \left[\begin{array}{c} \text{. account} \end{array} \right] \end{array} \right] \\ @ \left[\begin{array}{c} @ \left[\begin{array}{c} @ \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\text{detail} = \left[\begin{array}{c} 0 \\ 1 \\ 2 \end{array} \right]$$


In the above syntax:

fileset is one or more files, referenced by filename, group, and account. Default: all files in system.

detail is the type of information to be displayed. Default: Level 0 output (see below).

file.group.account Prints the file named in the group and the account designated.

file.group Prints the file named in the group designated.

file Prints the file named.

@.group.account Prints all files in the group named under the account designated.

@.group Prints all files for every such group in the system.

@.@.account Prints all files in all groups under the account named.

$\left. \begin{array}{l} @ \\ @.@ \\ @.@.@ \end{array} \right\}$ Prints all files in the system.

0 Requests display of file names only.

1 Requests display of the name, *ldn* and sector address of the first extent (i.e., file label) of a file.

2 Requests display of the name, creation date, last modification date, and last access date of a file.

NOTE

The @ designator may appear in any position—file, group, or account. Thus, *filename.@.account* and *filename.group.@* are legal entries. In this respect, PFIL does not demand the syntax constraints of the MPE :LISTF command. (See *MPE Commands Reference Manual* (part no. 30000-90009).)

PFIL

(Continued)

During a PFIL operation, if the high-order bit (bit 0) of the system switch register is set *ON*, the listing in progress is terminated and SADUTIL again issues the *ENTER NAME:* prompt.

If SADUTIL encounters an invalid file label during a PFIL operation, 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 under the account SYS ____

```
ENTER FUNCTION: PFIL
ENTER NAME: @.0.SYS
(Output follows.)
```

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

```
ENTER FUNCTION: PFIL
ENTER NAME: @,1
(Output follows)
```

PFRE

Prints the disc free-space map
for the specified disc volume.

SYNTAX

PFRE [*ldn*]

NOTES

The parameter *ldn* is the logical device number assigned to the volume.

You would use this function to determine whether to condense (COND) a volume. (See COND function.)

EXAMPLES

To print the disc free-space map pertaining to *ldn* 1 of the disc on which the volume is mounted, enter:

```
ENTER FUNCTION: PFRE 1
LDEV= 1, DRT= 15, UNIT= 0, TYPE= 0, SUBTYPE= 5
NO. OF ENTRIES: 5
TOTAL VOLUME CAPACITY: 19200 SECTORS
TOTAL FREE SPACE AVAILABLE: 1436 SECTORS
MAXIMUM CONTIGUOUS AREA: 1074 SECTORS
NO. ADDRESS SIZE
---
1. 13208 1
2. 13384 3
3. 14198 7
4. 14215 351
5. 18126 1074
ENTER FUNCTION:
```


PVOL

Prints information contained in the volume label of the specified disc volume.

SYNTAX

PVOL [*ldn*]

NOTES

The parameter *ldn* is the logical device number assigned to the volume. If the logical device number is omitted, then it is assumed to be 1 by default.

Note that the volume label is located in the first sector (sector 0) of the disc.

The volume label is written in this format:

Words	Contents
0-5	On the system disc, this field contains the bootstrap input/output program. On other discs, this field is filled with zeros.
6 (Bits 6:6) (Bits 12:4)	Disc type. Disc sub-type.
7	Cold-load count (incremented each time the system is cold-loaded).
8-9	The characters "3000," used to verify that the disc label is valid.
10-13	Volume name (Left-justified and padded with blanks).
14-127	Reserved (Initialized to zeros).

EXAMPLES

To print volume label information for *ldn* 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 disc and copies them to magnetic tape.

SYNTAX

SAVE

NOTES

This function is used in conjunction with the EDIT function (for disc edit) and the RECOVER2 program (for disc file creation) for recovering disc files following a catastrophic system failure. The SAVE function retrieves the files from disc and copies them to magnetic tape for later recovery. To perform this task, proceed as follows:

STEP	PROCEDURE
1	In response to the <i>ENTER FUNCTION:</i> prompt, enter <i>SAVE</i> .
2	SADUTIL now prints the prompt: MOUNT TAPE WITH WRITE RING ON MAG. TAPE UNIT 0 OF DRT 6 In response to this prompt, mount the tape to which the disc files are to be copied. (The program continues while you do this.)
3	SADUTIL prints the prompt: FILE NAME (OR LDEV #,%SECTOR ADDRESS)? Respond by indicating the file you wish copied to tape. This must be a file residing on a disc defined through the configuration portion of the SADUTIL/Operator Dialogue or through the CONF function. Indicate the file in any of the following three formats: a. By file name, in the format <i>filename.groupname.accountname</i> SADUTIL locates the specified file by searching the system file directory. Before copying the file, SADUTIL proceeds to Step 4. b. By the logical device number of the disc, followed by the sector address where you believe the file resides, in this format: <i>ldn, sectoraddress</i> If you plan to use this format, you should obtain a listing of file addresses via the PFIL function (<i>detail</i> = 1) prior to using the SAVE function, or from output previously obtained with the :STORE or :RESTORE command with SHOW option. You would normally use this format when you know that some of the information in the system file directory is no longer valid. When you use this format, SADUTIL transfers to Step 5 before copying the file specified. c. By specifying all files defined in the system file directory, as follows: @.@.@ With this format, the system file directory is used in locating the files. Before they are copied, SADUTIL transfers to Step 4. To terminate the SAVE function, enter a carriage return. In response, the <i>ENTER FUNCTION:</i> prompt is repeated.

SAVE

(Continued)

STEP	PROCEDURE
4	If either format <i>a</i> or <i>c</i> was selected under Step 3, SADUTIL prints this prompt: DATE? This prompt allows you to restrict retrieval to only those files modified since a specified date. Enter the date in this format: <i>mm/dd/yy</i> where <i>mm</i> = Two digits representing the month (for instance, 06 representing June). <i>dd</i> = Two digits representing the day of the month. <i>yy</i> = Last two digits of the year. SADUTIL now copies the files requested that were modified on or after the date specified. To omit the date specification and copy all files requested in Step 3, enter a carriage return. The following information is printed for all files copied: file name, group name, account name, logical device number as configured by SADUTIL, and disc sector address of file label. SADUTIL returns to Step 3 (if format <i>a</i> was selected) or terminates the SAVE function and prompts you to select a new function (if format <i>c</i> was selected). 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 <i>b</i> was selected under 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 the explanation—CONTENTS OF LABEL. Immediately after this information, SADUTIL prints the prompt: DO YOU WISH TO RETRIEVE THIS FILE(Y/N)? Respond by entering Y (for YES, to copy the file) or N (for NO, to avoid copying the file). In either case, SADUTIL performs the requested operation and returns to Step 3.

On the magnetic tape, each file is copied in blocks of 128-words, where the first block contains the file label, and the remaining blocks contain user labels (if any) and data. Each file is terminated by an end-of-file (EOF) mark. (A second EOF mark is written after the last file to denote the end-of-tape.) An individual file on tape cannot span more than one reel. During the

SAVE

(Continued)

SAVE operation, when the end-of-tape is sensed, the tape is back-spaced to the previous EOF mark and a second EOF mark (to denote the logical end-of-tape) is written. Then, SADUTIL prompts you to mount a new reel, and the file being written is copied in its entirety to this reel.

During the SAVE operation, the following words in Disc Sector 18 must be valid in order to locate the System Directory and perform range-checking on addresses referencing the Directory:

Word 14 (decimal): Directory Base Address

Word 13 (decimal): Directory size

If a disc error is detected when reading Sector 18, the following dialogue occurs after Step 1 in the above series of Steps:

STEP	PROCEDURE
1.1	CAN'T READ SECTOR 18 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 next operations in the order shown:

1. SADUTIL prints one of the following messages, explaining the error:

UNIT WENT OFF/LINE
NOT READY INTERRUPT
TRANSFER ERROR
CMD REJECT
TAPE RUN AWAY
TIMING ERROR
TAPE PARITY ERROR

2. SADUTIL then prints:

*TAPE { WRITE
EOF
BACKSPACE } ERROR*

3. Now, SADUTIL terminates the SAVE operation and prompts for a new function.

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

SAVE

(Continued)

EXAMPLES To retrieve the file SPL in the group PUB of the account SYS, enter SAVE and proceed as follows. (This example shows two methods.)

```
ENTER FUNCTION: SAVE
MOUNT TAPE WITH WRITE RING ON MAG. TAPE UNIT 0 OF DRT 6
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? SPL.PUB.SYS (By file name.)
DATE? return
SPL      .PUB      .SYS      1      %33560
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? 1,%33560
SPL      .PUB      .SYS      - CONTENTS OF LABEL (By disc address.)
RETRIEVE THIS FILE(Y/N)? N
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? return
ENTER FUNCTION
```

STOP

Terminates the SADUTIL program.

SYNTAX

STOP

NOTES

This function causes the SADUTIL program to execute a HALT instruction.

EXAMPLES

To terminate SADUTIL, enter:

```
ENTER FUNCTION: STOP
END OF PROGRAM.
```

ERROR MESSAGES

If an error occurs during the operation of SADUTIL, one of the messages shown below may appear on the Console. Following the message, SADUTIL operation continues and you are prompted for further input.

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.

DIRECTORY ERROR

During a COND operation, an error was found in the System File Directory. No condensing is performed. Use the EDIT function of SADUTIL to analyze the problem or see your System Manager.

DISC $\left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \\ \text{SEEK} \end{array} \right\}$ ERR ON LDEV# *ldn* STATUS = *%statusval* ADDR = *%sectoraddr*

SADUTIL was unable to complete the particular input/output operation to or from disc. In the message, *ldn* is the logical device number of the volume, *statusval* is the hardware status word of the device after the operation, and *sectoraddr* is the address of the disc sector where the error occurred. If the error occurs on a removable disc, remove the disc and replace it with a new scratch disc 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. (The account and group names, however, were valid and present.) Re-enter a correct file name.

FUNCTION NOT COMPLETED

The function in process encounters an irrecoverable error (such as a disc input/output error) and terminates. The operation requested was not achieved. This message typically follows an error message which explains the error.

GROUP NOT IN DIRECTORY

In a PFIL request, you referenced a file belonging to a group not present in the System File Directory. (The account name, however, 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 (such as EDIT, PFIL, SAVE, or STOP) not requiring such specification. Re-enter the function keyword correctly.

INVALID

You entered an invalid (unrecognizable) response to a prompt. Re-enter the response correctly when you are prompted again.

INVALID COMMAND

You entered an invalid EDIT command. Re-enter the correct command when you are prompted again.

INVALID DEVICE SPECIFICATION

You entered the desired function keyword (name) correctly but specified the input/output device incorrectly. For instance, you entered a special character or an unrecognizable number. When you are prompted again, re-enter the device specification correctly.

INVALID DISC ADDRESS

In the MODIFY command of the EDIT function, you specified an invalid disc address; *or*, a disc read or write operation attempted to access a sector address not on the volume. In the first case, specify a proper address in response to the next prompt; in the second case, a system problem exists and you should inform your System Manager.

LDEV *ldn* NOT DEFINED

You specified a logical device that has not been configured via the SADUTIL cold-load dialogue or the CONF function. Configure this device or specify another device.

LDEV *ldn* NOT READY

You requested a function that performs disc input/output, but the disc drive required is not ready (for instance, a removable cartridge has been removed). Ready the disc drive or reference another disc.

MISSING DEVICE SPECIFICATION FOR *volumename* VOLUME

The System Volume Table contains the name *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 dialogue when the system disc (*ldn* = 1) has not yet been configured. Proceed with the configuration.

VOLUME *volumename* FOR LDEV *ldev* NOT IN VOLUME TABLE

This message appears when a configured disc is not represented in the System Volume Table. The message only appears during those functions that require all volumes to be mounted (such as COND, PFIL, and SAVE).

RECOVER2

SECTION

II

RECOVER2 allows you to re-create, in the system, disc files that have been copied to tape by the SADUTIL program (Section I).

NOTE

RECOVER2 runs in Privileged Mode only; you must have Privileged Mode Capability to use it.

To use RECOVER2, follow the directions in table 2-1, below:

Table 2-1. Using RECOVER2

STEP	PROCEDURE
1	Following system termination, retrieve all necessary files from disc, using the SAVE function of SADUTIL (as defined in Section I).
2	Re-initiate MPE through the RELOAD option of the MPE Initiator program, as directed in the <i>Console Operator's Guide</i> (part no. 30000-90013).
3	If the magnetic tape used for RELOAD does not contain files that completely reflect the previously-existing account structure, recreate this structure using the appropriate commands described in the <i>System Manager/System Supervisor Manual</i> (part no. 30000-90014).
4	Mount the tape prepared by SADUTIL that contains the files to be copied back into the system, and ready the tape unit.
5	Log-on to the system under the user and account names MANAGER.SYS. Then, run RECOVER2 as follows: :RUN RECOVER2.PUB.SYS RECOVER2 begins execution, reading the files from the tape and copying them to disc.
6	If RECOVER2 encounters one or more files on the tape that already exist in the system, it prompts you (<i>only once</i>) with the message: WISH TO KEEP EXISTING COPIES OF FILES? (Y/N) Enter Y (for YES, to retain the files as they exist) or N (for NO, to overwrite the disc files with the files from tape).
7	When all files have been copied from the tape, RECOVER2 prints the names of those files and prompts you for any additional tape containing files to be copied back into the system: IS THERE ANOTHER RECOVERY TAPE (Y/N)?

8	Enter <i>Y</i> (for YES, to specify another tape) and mount the tape, or <i>N</i> (for NO, to terminate the program). If you enter <i>Y</i>, RECOVER2 reads the files as in Step 5 and continues. If you enter <i>N</i>, RECOVER2 prints the following message and terminates: END OF PROGRAM
---	--

NOTE

When RECOVER2 terminates, you can restore all files to the most current level by using the :RESTORE command with the KEEP keyword parameter. (See *System Manager/System Supervisor Manual*.)

DPAN2

SECTION

III

DPAN2 is a program that produces an analytical formatted listing of main memory contents based on a memory dump taken after system termination. This listing can then be studied for general systems analysis or to determine the cause of any failure.

NOTE

No special MPE capabilities are required to run this program.



The DPAN2 listing is obtained as follows:

1. Immediately after system termination, request a main-memory dump. This dump is copied onto magnetic tape through special microcode. It provides an instant "snap shot" picture of memory at the time of the termination.
2. When the system has been re-started, run the program DPAN2 to produce the formatted listing, using the memory dump on tape as input.

It is highly recommended that these steps be performed any time the system terminates for an unknown reason.

To obtain the memory dump, follow the steps in table 3-1, below.

Table 3-1. Obtaining Memory Dump

STEP	PROCEDURE
1	If the central processor has not already halted at the time you wish to produce the dump, halt it by pressing the RUN/HALT switch on the system control panel.
2	Mount and load a reel of tape, with the write-enable ring attached, on the magnetic tape unit and press the 0 switch. (The tape unit used is normally the one configured as DRT Entry No. 6. See your System Manager for further details.) Then, turn the unit ON LINE. Make certain that the tape you use is clean and non-defective.
3	Press the DUMP and ENABLE switches on the system control panel simultaneously, and watch the magnetic tape and current instruction register (CIR). (The CIR denotes which section of memory is being dumped; when the dump is concluded, the CIR indicates how much memory has been written to tape.) The tape should wind forward several feet and the central processor should halt without rewinding the tape. If this occurs, and the CIR contains the value shown in table 3-2 for the memory size of your system, the dump has succeeded. Re-start the system and prepare to run DPAN2 as directed below. If the central processor did not halt, the dump is invalid; attempt another dump using a new tape. Some of the reasons for unsuccessful dumping are: tape unit not ON-LINE and ready; tape unit 0 not selected; tape reel missing write-enable ring; defective magnetic tape which causes a read-after-write error; memory parity error. When a dump is retaken, the S-bank and CIR contents are invalid; approximately 30 memory locations following the DRT entry for the dump tape unit are also invalid.

The proper CIR contents following a dump for each memory size available on the Series II are listed in table 3-2. The uppermost (rightmost) three bits in the CIR denote the number of banks. When you visualize the CIR contents with the rightmost three bits removed and appended to the left end of the register, you obtain the amount of memory dumped.

Table 3-2. CIR Contents After Dump

MEMORY SIZE (in words)	CIR CONTENTS (in octal)
64K	000001
96K	100001
128K	000002
160K	100002
192K	000003
224K	100003
256K	000004

To obtain the formatted listing, proceed as directed in table 3-3.

Table 3-3. Obtaining DPAN2 Listing

STEP	PROCEDURE
1	Re-start the system via the COLD LOAD, COOLSTART, WARMSTART, or RELOAD options of the system re-start program, Initiator.
2	Log-on to the system. Then, run DPAN2 by entering: <u>:RUN DPAN2.PUB.SYS</u> NOTE DPAN2 first attempts to open an input file named MDUMP.PUB.SYS on disc. If this attempt does not succeed, then DPAN2 opens a file whose formal designator is MDUMP, with the default device class name TAPE. DPAN2 directs its output to a file whose formal designator is DPANLIST; by default, this is any file with the device class name LP (if DPAN2 is run in interactive mode) or the actual designator \$STDLIST (if DPAN2 is run in batch mode). If you wish to modify these file characteristics, you must enter MPE :FILE commands prior to running DPAN2. When DPAN2 enters execution, it opens all files required. When DPAN2 attempts to read the memory dump from tape, a request is printed on the system console to mount the tape (MDUMP). The system waits for the tape to be mounted. DPAN2 then reads the memory dump and re-winds the tape reel. Then, DPAN2 produces the formatted listing.
3	Remove the magnetic tape from the tape unit.

Table 3-3. Obtaining DPAN2 Listing (Continued)

STEP	PROCEDURE
4	Remove the listing. The listing will contain the following information, retrieved at the time of system termination: identification page, register pages, fixed locations in low memory, code segment table (CST), code segment table extension (CSTX), data segment table (DST), process control block (PCB), monitor table, working set table, memory table, virtual disc space page allocation information, virtual disc space bit map, device reference table (DRT), interrupt linkage table, logical/physical device table (LPDT), device information table (DIT), input/output request table, system buffer information, terminal buffer contents, timer request list, main memory contents (in octal), and index.

LISTLOG2

SECTION

IV

LISTLOG2 prints, in ASCII format, the contents of any MPE Log Files existing in the system. These files describe system usage on a session/job basis, showing a running log of events such as session/job initiation and termination, process termination, file closing, and system shut-down. These files are created under the user and account names MANAGER.SYS. For more information about Log Files, see the *System Manager/System Supervisor Manual* (part no. 30000-90014).

NOTE

To use this program, you must have System Manager Capability and must log-on under the user/account names MANAGER.SYS.

The output from LISTLOG2 can be directed to any output file on any device. (LISTLOG2 recognizes this file by the formal designator LOGLIST, assigned the default device class name LP.) LISTLOG2 allows you to specify the type of log records to be displayed. To use LISTLOG2, follow the directions in table 4-1.

Table 4-1. Obtaining LISTLOG2 Output

STEP	PROCEDURE
1	Log-on to the system under MANAGER.SYS and run LISTLOG2 by entering: :<u>RUN LISTLOG2.PUB.SYS</u>
2	LISTLOG2 responds by printing a heading identifying the program, followed by the prompt: ENTER FIRST AND LAST LOG FILES TO BE ANALYZED FIRST? Enter the identifying number of the first Log File to be printed.
3	LISTLOG2 then prints the prompt: LAST? Enter the number of the last Log File to be printed. If the file indicated in Step 2 is the <i>only</i> file to be printed, enter <i>return</i> in Step 3; in this case, only the first log file will be analyzed and printed.
4	LISTLOG2 now prints another prompt: ENTER EVENTS TO BE PRINTED TYPE NO. EVENT 0 LOG FAILURE 1 SYSTEM UP 2 JOB INITIATION 3 JOB TERMINATION 4 PROCESS TERMINATION

Table 4-1. Obtaining LISTLOG2 Output (Continued)

STEP	PROCEDURE
4 (Continued)	5 FILE CLOSE 6 SYSTEM SHUTDOWN 7 POWER FAILURE 8 SPOOLING LOG RECORD 9 LINE DISCONNECTION 10 LINE CLOSE 11 I/O ERRORS ENTER EVENT NUMBERS SEPARATED BY COMMAS A CARRIAGE RETURN ASSUMES ALL EVENTS WILL BE EVALUATED Enter the logging events (records) you want printed, selected by number from the above list. (For example, to display job-initiation records, enter 2.) If you select more than one event, separate the event numbers by commas. (For instance, for the first three events from the above list, enter 0, 1, 2.) To request printing of all events, enter <i>return</i>.
5	LISTLOG2 now prints: DO YOU WANT TO PURGE LOG FILES? Enter Y (for YES, to purge all log files listed) or N (for NO, to retain these files).
6	LISTLOG2 now prints: DO YOU WISH TO RUN AGAIN(Y OR N)? Enter Y (for YES, to run LISTLOG2 again) or N (for NO, to terminate LISTLOG2).
7	If you enter <i>N</i> in Step 6 above, LISTLOG2 prints the following message and terminates: END OF PROGRAM

An example of a LISTLOG2 run appears in figure 4-1. In this figure, user input is *underlined* for clarity.

```

:RUN LISTLOG2.PUB.SYS

LIST LOG FILE PROGRAM VERSION 00.00 4/1/76
ENTER FIRST AND LAST LOG FILE TO BE ANALYZED
FIRST?2824
LAST?2825
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
ENTER EVENT NUMBERS SEPARATED BY COMMAS
A CARRIAGE RETURN ASSUMES ALL EVENTS WILL BE EVALUATED
11
DO YOU WANT TO PURGE LOG FILES?YES

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

END OF PROGRAM

```

Figure 4-1. LISTLOG2 Dialogue

If any errors occur during your dialogue with LISTLOG2, the appropriate error messages are printed on the standard listing device. Examples of such messages are:

FOPEN ERROR TO LOGFILE *xxxx* ERROR NUMBER *yy*

FWRITE FAILED TO LINE PRINTER ERROR NUMBER *yy*

In these messages, *xxxx* is the log file number; *yy* is the FCHECK *errorcode*, as described in *MPE Intrinsic Reference Manual*.

Fatal errors call the QUIT intrinsic (Intrinsic No. 99) after the error message is printed, terminating the program.

LISTEQ2

SECTION

V

LISTEQ2 lists the MPE :FILE commands used, and temporary files created, during the current session/job. To accomplish this, LISTEQ2 accesses the job directory tables (JDTs).

NOTE

No special MPE capabilities are required to run this program.

To use LISTEQ2, log-on to the system and enter:

```
:RUN LISTEQ2.PUB.SYS
```

By default, LISTEQ2 writes all output to the standard output file (\$STDLIST). You may, however, re-direct the output to another file by using a :FILE command referencing the formal file designator LIST and specifying ;PARM=1 in the command to run LISTEQ2, as follows:

```
:FILE LIST; DEV=LP          } Directs output to any device with class  
:RUN LISTEQ2.PUB.SYS; PARM=1 } name LP (normally, a line printer).
```

When no :FILE commands or session/job temporary files exist for the current session/job, LISTEQ2 prints a message indicating this.

An example of a LISTEQ2 run appears in figure 5-1.

```
:FILE TAPE; DEV=TAPE  
:FILE PRINT; DEV=LP  
:BUILD INPUT; REC=40, 3, F, ASCII; TEMP  
:RUN LISTEQ2.PUB.SYS  
LISTEQ          (CU.07)  
  
***TEMP FILES  
  
INPUT.PUB.TECHPUBS  
  
***FILE EQUATIONS  
  
:FILE TAPE;DEV=TAPE  
:FILE PRINT;DEV=LP  
  
END OF PROGRAM  
:
```

Figure 5-1. LISTEQ2 Input/Output

DISKED2 allows you to modify or display the contents of a disc according to absolute sector addresses.

NOTE

To execute DISKED2, you must have either the System Manager or System Supervisor capability. This program runs in Privileged Mode.

WARNING

The normal checks and limitations that apply to the standard users in MPE are bypassed in privileged mode. It is possible that modification of privileged data on disc can destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged data. This service, which is not provided under the standard service contract, is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.

To run DISKED2, log-on to the system and enter:

```
:RUN DISKED2.PUB.SYS
```

The program identifies itself by printing the following heading:

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

Then, the program prompts you for a command by printing a greater-than sign (>). Respond by entering one of the commands shown in table 6-1. These commands all permit free-format parameters. When you enter an octal number as a parameter, indicate this by preceding the value with a percent sign (%). After the command is executed, the prompt is repeated.

Table 6-1. DISKED2 Commands

COMMAND	FUNCTION
BASE <i>abssector</i>	Specifies the absolute sector number for the DUMP and MODIFY commands, discussed below.
DEBUG	Calls the MPE Debug subsystem.
DISC <i>ldev</i>	Specifies the disc to be modified or listed, by logical device number (<i>ldev</i>). Initial setting: 1.

Table 6-1. DISKED2 Commands (Continued)

COMMAND	FUNCTION
DUMP [<i>relsector</i>] [, <i>numsectors</i>]	Displays selected sectors on list device. The starting sector is determined by adding <i>relsector</i> (the relative sector number) to the <i>abssector</i> (absolute sector number) entered in the BASE command. Defaults: <i>relsector</i> = 0, and <i>numsectors</i> (number of sectors) = 1. Initial setting: <i>abssectors</i> = 0.
EXIT	Terminates the program.
HELP	Displays a list of commands on \$STDLIST (standard listing device). Used to refresh your memory on commands available and their formats.
LIST $\left[\begin{array}{l} \textit{ldev} \\ \textit{devclass} \end{array} \right]$	Specifies output device for DUMP command listing output, by logical device number (<i>ldev</i>) or device class name (<i>devclass</i>). Initial setting: \$STDLIST. Default: \$STDLIST.
MODIFY <i>sectornum</i> , <i>relwordaddr</i> [, <i>numwords</i>]	Specifies data to be modified. In this command, <i>sectornum</i> is the absolute sector address (decimal or octal) of the data. The address you actually modify is calculated by adding <i>abssector</i> from the BASE command to <i>sectornum</i> as specified in MODIFY. The <i>relwordaddr</i> parameter is the address of the first word to be modified, relative to Word 0 of <i>sectornum</i>. The data may span sectors. The <i>numwords</i> parameter is the total number of words to be modified. Default: 1. The MODIFY command prompts you with the number of the sector being modified, for each word to be changed: <pre>SECTOR % <i>sectornum</i> LDEV= %<i>ldev</i> <i>relwordaddr</i>: <i>wordcontents</i>%</pre> In this message, <i>sectornum</i> is the absolute sector address, <i>ldev</i> is the logical device number, <i>relwordaddr</i> is the address of the word to be modified, and <i>wordcontents</i> is the present contents of that word. Respond to this prompt by entering a <i>new value</i> to replace the data, an asterisk (*) or <i>return</i> to rewrite the current value, or a <i>slash mark</i> (/) to abort the MODIFY operation. After each sector is actually written, the following message appears: WRITTEN

Table 6-1. DISKED2 Commands (Continued)

COMMAND	FUNCTION
WIDTH	Specifies that ASCII output from the DUMP command appears beside or below the octal dump output. Before this information is printed, the following prompt appears: NARROW FORMAT? To place the ASCII information below the octal dump, enter Y. To place this information at the right of the octal information, enter any other response. This command writes records that are one character less than the configured record width of the listing device. Output characters at and beyond the record width are truncated.

If an error occurs during your interaction with DISKED2, the program prints one of the messages described in table 6-2, followed by the indication *ERROR on the next line.

Table 6-2. DISKED2 Error Messages

MESSAGE	MEANING
FWRITE ERR ON LIST <i>fchecknum</i>	Output from the program to the listing device was not completed. The parameter <i>fchecknum</i> lists the error code returned by the FCHECK intrinsic, described in the <i>MPE Ininsics Reference Manual</i> (part no. 30000-90010).
[DISC <i>attiono</i>	The ATTACHIO routine failed on the disc; <i>attiono</i> is an internal error number returned by this routine.
INVALID DISC ADDRESS	Address specified by BASE command 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.
DEVICE NOT DISC	A command referenced a device that is not a disc.
UNABLE TO OPEN LIST DEVICE	DISKED2 could not open the list device.

FREE2 prints histograms of available disc space and the total number of entries in the disc Free-Space Table. In the listing, each disc configured in the system is identified by volume name, and the histogram for that disc follows this name. The total number of Free-Space Table entries in the system appears at the end of the list. This output aids in determining how a disc's resources are being used and in estimating the degree of fragmentation of free space.

NOTE

No special MPE capabilities are required to run this program.

To run FREE2, log-on to the system and enter:

```
:RUN FREE2.PUB.SYS
```

An example of FREE2 input/output appears in figure 7-1. All numbers output are decimal values.

*RUN FREE2.PUB.SYS

VOLUME= M7905

LARGEST FREE AREA= 15114

SIZE	COUNT	SPACE	AVERAGE
>100000	0	0	0
>10000	1	15114	15114
>1000	0	0	0
>100	1	126	126
>10	2	78	39
>1	0	0	0

TOTAL FREE SPACE=15318

VOLUME= M26880

LARGEST FREE AREA= 15991

SIZE	COUNT	SPACE	AVERAGE
>100000	0	0	0
>10000	1	15991	15991
>1000	0	0	0
>100	11	3133	284
>10	111	3127	28
>1	271	1065	3

TOTAL FREE SPACE=23316

VOLUME= M26881

LARGEST FREE AREA= 44861

SIZE	COUNT	SPACE	AVERAGE
>100000	0	0	0
>10000	1	44861	44861
>1000	3	6229	2076
>100	23	7454	324
>10	93	2036	21
>1	224	830	3

TOTAL FREE SPACE=61410

⋮

SYSTEM TOTAL FREE SPACE=143782

END OF PROGRAM

Volume name.

Largest free area (in sectors).

size > 100 means that this line contains statistics on all free spaces with sizes greater than 100 sectors (but not exceeding 1000 sectors).

count = No. of free spaces of this size range.

Space = sum of no. of sectors in all free spaces of this size range.

Average = Average size (in sectors) of all free spaces in this size range.

Total free spaces on this volume (in sectors).

Total of all free spaces on all volumes (in sectors).

Figure 7-1. FREE2 Input/Output

LISTDIR2

SECTION

VIII

LISTDIR2 displays the attributes of specified accounts, groups, users, and files. This program provides information similar to that provided by the :LISTACCT command, available to System Managers and Supervisors; the :LISTGROUP and :LISTUSER commands, available to System Managers and Supervisors, and Account Managers; and the :LISTF command, available in some form to all users.

NOTE

No special MPE capabilities are required to run this program, although a lack of certain capabilities (such as System Manager or Account Manager) may restrict the extent of the information you can list.

To run LISTDIR2, log-on to MPE and enter:

```
:RUN LISTDIR2.PUB.SYS
```



The program responds by printing the following message on your standard listing device, and then prompting you for commands by printing a greater-than sign (>):

```
LISTDIR2
```

```
TYPE 'HELP' FOR AID  
>
```

COMMANDS

The LISTDIR2 commands allow you to:

- List the attributes of accounts, users, groups, and files.
- List the security provisions for one or more files.
- List the syntax rules for all LISTDIR2 commands.

You enter LISTDIR2 commands through the standard input device (\$STDIN). LISTDIR2 directs prompts and error messages to the standard listing device (\$STDLIST). LISTDIR2 also directs its formatted output to \$STDLIST, unless you specify otherwise, through the formal file designator OUT. To override this default and direct this output to another device, you must enter a :FILE command equating OUT to the desired device and then specify the keyword parameter group ;*PARAM*=1 in the :RUN command, as follows:

```
:FILE OUT; DEV=LP
```

Specifies Line Printer for output.

```
:RUN LISTDIR2.PUB.SYS; PARAM=1
```

Runs LISTDIR2.

Formatted output for most individual LISTDIR2 commands can also be re-directed through the *listfile* parameter provided with these commands.

Each command consists of a command name followed in most cases by an optional parameter list. The command name must begin in the first column of the record on which it appears (in batch job mode) or immediately following the prompt character (in interactive session mode). Blanks may appear anywhere between syntactic elements such as names, separators, or keywords. However, a qualified identity such as *GROUP.ACCOUNT* may *not* contain embedded blanks.

Where the *listfile* parameter appears in a command, it indicates the file to which output produced by the command is written. This parameter can name any valid file designator. If this designator is a back-reference to a :FILE command, the designator must be preceded by an asterisk (*). Otherwise, the designator must reference an existing session/job temporary or permanent file.

Where the *PASS* parameter is used, this indicates that normally-restricted information such as passwords, file lockwords, creator identities, and disc addresses are to be displayed. If this parameter is omitted, or if you do not have sufficient capability to list this information, a double asterisk (**) appears in place of the restricted information.

At any time before you press the *return* key to terminate a command entry in session mode and transmit the command to the system, you can suspend LISTDIR2 by pressing the *break* key. (On some terminals, the *break* key is labeled *INTERRUPT* or *ATTENTION*.) This places the terminal in system break mode, transferring control to the MPE command interpreter program. If you wish to return control to LISTDIR2, you can enter the MPE command :RESUME or re-run the program by entering the :RUN command again.

NOTE

You cannot enter system break mode *during* execution of a LISTDIR2 command.

If you wish to halt the execution of a LISTDIR2 command, enter Y^c (CONTROL-Y). When command execution halts, LISTDIR2 prompts you for another command.

In general, the MPE security provisions restrict your access to certain information. For example, if you have only standard user capabilities, you may list the attributes of your log-on account, group, and user-name definition only. Generally, you may list the attributes of any file (although display of some attributes may be restricted). If you have Account Manager capability, you may list the attributes of other groups and user-name definitions within your account and display normally-restricted data such as passwords, lockwords, and file-creator identities in this account. If you have System Manager capability, you may display these elements for other accounts as well. For further information, see *Security Restrictions* later in this section.

The LISTDIR2 commands are presented in table 8-1.

Table 8-1. LISTDIR2 Commands

COMMAND	FUNCTION
LISTACCT $\left[\begin{array}{c} @ \\ \text{accountname} \end{array} \right] [, \text{listfile}] [; \text{PASS}]$	Lists attributes of the specified account, such as capabilities and security. In the parameter list, <i>accountname</i> is the name of the account and @ indicates <i>all</i> accounts in the system. If these parameters are omitted, the properties of the log-on account are listed. You may specify an account other than your log-on account and may list the account password only if you have System Manager Capability.
$\text{LISTF} \left\{ \begin{array}{l} \text{filename} [, \text{groupname} [, \text{accountname}]] \\ @ \left[\begin{array}{l} \text{groupname} [, \text{accountname}] \\ .@ \left[\begin{array}{l} .@ \\ \text{accountname} \end{array} \right] \end{array} \right] \end{array} \right\}$ [, listfile] [; PASS] [; MAP]	Lists attributes of a specified file or set of files. These properties include the number of sectors used, creation date, disc unit number, file label address, and file security. For program files, program capabilities and the number of segments are also listed. In the parameter list, you <i>must</i> specify at least one file. The symbol @ indicates <i>all</i>; for instance, @.GROUPA.ACCOUNTA lists all files in GROUPA of ACCOUNTA; @.@ACCOUNTA lists all files in all groups of ACCOUNTA; @.@.@ lists all files in the system (all files in all groups of all accounts). If you have standard user capabilities only, you cannot specify @.@, @.@.accountname, or @.@.@. If your highest capability is Account Manager, you cannot specify @.@.accountname (for an account other than your log-on account) or @.@.@. The MAP and PASS parameters may appear in any order. MAP provides a list of the disc addresses for each extent; this feature is only available to the creator of the file and to System and Account Managers. LISTF has the same effect as the MPE command :LISTF fileset, -1.
LISTGROUP $\left[\begin{array}{c} \text{groupname} [, \text{accountname}] \\ @ \left[\begin{array}{c} .@ \\ \text{accountname} \end{array} \right] \end{array} \right]$ [, listfile] [; PASS]	Lists attributes of a specified group or set of groups. If no group is specified, the properties of the log-on group are listed. An Account Manager may specify any group in his log-on account. A System Manager may specify any group in the system. Only System or Account Managers may display group passwords.
LISTUSER $\left[\begin{array}{c} \text{username} [, \text{accountname}] \\ @ \left[\begin{array}{c} .@ \\ \text{accountname} \end{array} \right] \end{array} \right]$ [, listfile] [; PASS]	Lists attributes of a specified user or set of users, such as capabilities and home group. If no user is specified, the properties of your log-on user-name definition are listed. When you specify the log-on user name, the session/job number, terminal logical device number, and log-on group name are also listed. An Account Manager may specify any user name in the log-on account. A System Manager may specify any user name in the system. Only system or Account Managers may display user passwords.

Table 8-1. LISTDIR2 Commands (Continued)

COMMAND	FUNCTION
LISTSEC { filename [.groupname [.accountname]] @ [.groupname [.accountname]] .@ [.@ [.accountname]] } [,listfile] [;PASS]	Lists total security provisions for specified files. This includes account, group, and file-level security as well as effective log-on file security for you (access available to you). The restrictions on the output available with this command are similar to those for LISTF. You must specify at least one file in this command.
HELP	Lists a summary of the syntax rules for the LISTDIR2 commands.
EXIT	Terminates the LISTDIR2 program.

Examples of listings produced by the LISTDIR2 commands appear in figure 8-1. In these listings, certain abbreviations appear whose full meanings may not be initially obvious. Among these are *accounting*, *group*, *user*, and *program file capabilities* (defined for your convenience in table 8-2) and *file security access-mode/user-type information* (defined in tables 8-3 and 8-4). Further information about these items appears in the *MPE Commands Reference Manual* (part no. 30000-90009). In addition, the abbreviations *S*, *B*, and *W* indicate *sectors*, *bytes*, and *words* respectively. The term *COLD-LOAD I.D.* refers to a count incremented each time the system is cold loaded from disc; it helps identify the status of files in the system whenever a dump is taken. The notations *LABELS* and *MAX LABELS* refer to user-written file labels, not standard labels written by MPE. *SEC OFFSET* refers to the number of sectors within a file, counting from the sector containing the file label (Sector 0). *FLAGS* indicates if and why a file has been locked down for exclusive access by a program. The terms *FCB VECTOR* and *INX PTR* refer to internal information useful only to systems analysts as aids in debugging.

>LISTACCT TECHPUBS

ACCOUNT: TECHPUBS

DISC SPACE: 1882(S)
CPU TIME: 3(SEC)
CONNECT TIME: 7(MIN)
DISC LIMIT: 2000(S)
CPU LIMIT: UNLIMITED
CONNECT LIMIT: UNLIMITED
MAX PRI: 150
GRP INX PTR: %545
USR INX PTR: %544
CAP: AM,AL,GL,ND,SF,IA,BA

PASSWORD: **
LOC ATTR: %0
SECURITY--READ: AC
WRITE: AC
APPEND: AC
LOCK: AC
EXECUTE: AC

>LISTGROUP @.TECHPUBS

GROUP: PUB.TECHPUBS

DISC SPACE: 1882(S)
CPU TIME: 3(SEC)
CONNECT TIME: 7(MIN)
DISC LIMIT: 2000(S)
CPU LIMIT: UNLIMITED
CONNECT LIMIT: UNLIMITED
FILE INX PTR: %546
CAP: IA,BA

PASSWORD: **
SECURITY--READ: ANY
WRITE: AL,GU
APPEND: AL,GU
LOCK: AL,GU
EXECUTE: ANY
SAVE: AL,GU

⋮

>LISTUSER MAC; PASS

USER: MAC.TECHPUBS

HOME GROUP: PUB
MAX PRI: 150
LOGON CNT: 1
CAP: AM,AL,GL,ND,SF,IA,BA

PASSWORD:
LOC ATTR: %0

LOGON GROUP: PUB
LOGON DEV #: 48

SESSION #: S109

Figure 8-1. LISTDIR2 Command Examples (Sheet 1 of 3)

>LISTF MYPROGX.PUB.TECHPUBS

FILE: MYPROGX.PUB.TECHPUBS

FCODE: PROG	FOPTIONS: BINARY, FIXED
BLK FACTOR: 1	CREATOR: **
REC SIZE: 256(B)	LOCKWORD: **
BLK SIZE: 128(W)	SECURITY--READ: ANY
EXT SIZE: 6(S)	WRITE: ANY
# REC: 5	APPEND: ANY
# SEC: 6	LOCK: ANY
# EXT: 1	EXECUTE: ANY
MAX REC: 5	**SECURITY IS ON
MAX EXT: 1	COLD LOAD ID: %12542
# LABELS: 0	CREATED: TUE, 27 JAN 1976
MAX LABELS: 0	MODIFIED: TUE, 27 JAN 1976
DISC DEV #: 18	ACCESSED: TUE, 27 JAN 1976
DISC TYPE: 0	LABEL ADR: **
DISC SUBTYPE: 3	SEC OFFSET: %1
CLASS: DISC	FLAGS: NO ACCESS
FCB VECTOR: %0	
# SEG: 1	TOTAL DB: %2
STACK: %1440	DL: %0
MAXDATA: DEFAULT	CAP: IA,BA

⋮

>LISTSEC MYPROGX.PUB.TECHPUBS

FILE: MYPROGX.PUB.TECHPUBS

SECURITY--READ: AC
(ACCT) WRITE: AC
APPEND: AC
LOCK: AC
EXECUTE: AC

SECURITY--READ: ANY
(GROUP) WRITE: AL,GU
APPEND: AL,GU
LOCK: AL,GU
EXECUTE: ANY
SAVE: AL,GU

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

FOR MAC.TECHPUBS: READ,WRITE,APPEND,LOCK,EXECUTE

Figure 8-1. LISTDIR2 Command Examples (Sheet 2 of 3)

>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>]
EXIT
```

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

CONTINUE? (Y/N)Y

EXAMPLES: LISTF MYFILE.MYGROUP.MYACCT
LISTF @.MYGROUP
<LISTFILE> IS ANY VALID FILE DESIGNATOR, PRECEDED BY AN ASTERISK TO BACK-REFERENCE A :FILE COMMAND. WITHOUT AN ASTERISK, THE FILE DESIGNATOR MUST BE AN EXISTING TEMPORARY OR PERMANENT FILE.
EXAMPLE: LISTACCT @, *PRINTER

"PASS" PERMITS ACCOUNT AND SYSTEM MANAGERS TO DISPLAY PASSWORDS, LOCKWORDS, CREATOR ID'S, AND FILE DISC ADDRESSES. THE CREATOR OF A FILE IS ALSO PERMITTED TO DISPLAY DISC ADDRESSES AND THE CREATOR ID.

"MAP" PERMITS THE FILE CREATOR AND ACCOUNT AND SYSTEM MANAGERS TO DISPLAY THE FILE EXTENT MAP.

>EXIT

END OF PROGRAM

:

Figure 8-1. LISTDIR2 Command Examples (Sheet 3 of 3)

Table 8-2. Capability-Set Abbreviations

ABBREVIATION	MEANING
<i>User Attributes:</i> SM AM AL GL DI OP	System Manager Account Manager Account Librarian Group Librarian Diagnostician System Supervisor
<i>File-Access Attributes</i> SF ND	Permanent Files Access of non-sharable I/O devices
<i>Capability-Class Attributes:</i> PH DS MR PM IA BA	Process-Handling Extra Data Segments Multiple RINS Privileged Mode Interactive Access Local Batch Access } Standard Capabilities

Table 8-3. Access-Mode Abbreviations

ABBREVIATION	ACCESS-MODE	MEANING
R	Reading	Allows users to read files.
L	Locking	Permits a user to prevent concurrent access to a file by himself and another user, thus allowing accessing user to <i>lock</i> file while he is using it. Specifically, it permits use of the FLOCK and FUNLOCK intrinsics, and the exclusive-access option of the FOPEN intrinsic. See the <i>MPE Intrinsics Reference Manual</i> (part no. 30000-90010).
A	Appending	Allows users to add information and disc extents to files, but prohibits them from altering or deleting information already written. This access mode implicitly allows the locking (L) access mode described above.
W	Writing	Allows users general writing access, permitting them to add to, delete, or change any information on files. This includes removing entire files from the system with the :PURGE command. Writing access also implicitly allows the locking (L) and appending (A) access modes described above.

Table 8-3. Access-Mode Abbreviations (Continued)

ABBREVIATION	ACCESS-MODE	MEANING
S	Saving	Allows users to declare files <i>within their groups</i> permanent (via the :SAVE command or FCLOSE intrinsic), and to rename such files (via the :RENAME command). This ability includes the creation of a new permanent file with the :BUILD command.
X	Executing	Allows users to run programs stored on files, with the :RUN command or CREATE intrinsic.

Table 8-4. User-Type Abbreviations

ABBREVIATION	USER/TYPE	MEANING
ANY	Any User	Any user defined in the system; this includes all categories defined below.
AL	Account Librarian User	User with Account Librarian Capability, who can manage certain files within his account that may or may not all belong to one group.
GL	Group Librarian User	User with Group Librarian Capability, who can manage certain files within his home group.
CR	Creating User	The user who created this file.
GU	Group User	Any user allowed to access this group as his log-on or home group, including all GL users applicable to this group.
AC	Account Member	Any user authorized access to the system under this account; this includes all AL, GU, GL, and CR users under this account.

SECURITY RESTRICTIONS

Access to information through the LISTDIR2 commands is restricted. These restrictions are derived from MPE to prevent indiscriminate disclosure of group and account names and passwords, and prevent non-responsible users from tying up system resources:

A System Manager has unrestricted access to the LISTDIR2 output information.

An Account Manager may specify any group or user name in his log-on account.

A standard user (one who is neither a System Manager nor an Account Manager) may specify only his log-on account, group, and user name.

Passwords, lockwords, creator identities, file label addresses, and privileged file codes are displayed only when PASS is specified and only for qualified users (as noted below).

Account passwords may be listed only by System Managers.

Group and user passwords may be listed only by an Account Manager or System Manager.

File lockwords and creator identities may be listed only by an Account Manager or System Manager.

Disc file addresses and extent maps may be displayed by the creator of the file, or an Account Manager or System Manager.

Privileged file codes may be displayed by an Account Manager, System Manager or the creator of the file (if creator has Privileged Mode Capability).

The file sets *@.@* and *@.@.accountname* may be specified only by an Account Manager (where *accountname* indicates the log-on account) or by a System Manager.

The file set *@.@.@* may be specified only by a System Manager.

PATCH

SECTION

IX

PATCH allows you to display or modify the current contents of a program file (without recompilation) by directly accessing the object code in the file. You may wish to use this method to modify a program at the instruction level if the modification is simple and can be readily inserted into the sequence of instructions already in the file.

NOTE

No special MPE capabilities are required to run this program.

WARNING

This program can be used to modify the contents of privileged program files, in which case the normal checks and limitations that apply to the standard users in MPE are bypassed. It is possible that modification of privileged software can destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged software. This service which is not provided under the standard service contract, is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.

The system manager must ensure that users are not allowed access to any privileged program files. He should restrict the use of PATCH only to necessary users, perhaps via a file lockword.

To use PATCH, follow the directions in table 9-1.

Table 9-1. Using PATCH

STEP	PROCEDURE
1	Log-on to the system. Then run PATCH by entering: :RUN PATCH.PUB.SYS
2	PATCH responds by printing a heading identifying itself, followed by a prompt for the name of the file you wish to modify or list. PROGRAM PATCHER CU.06 FILE=? Enter the name of the appropriate program file.
3	PATCH now prints a question mark (?) as a prompt for a PATCH command: ? To display a code segment, enter the D command, in the following format: <i>D,segment-number, address [,number-of-locations]</i>

Table 9-1. Using PATCH (Continued)

STEP	PROCEDURE
3 (Continued)	To modify a code segment, enter the M command as follows: M,segment-number, address [,number-of-locations] To display the global area of the initial stack, enter: D,address [,number-of-locations] To modify the global area of the initial stack, enter: M,address [,number-of-locations] In these commands, all numbers and addresses entered are interpreted as <i>octal</i> values. The contents of the specified addresses are displayed, as shown in the example in figure 9-1. If you omit the <i>number-of-locations</i> parameter, one location is assumed. To enter a new value during a Modify (M) operation, type the new value after the value displayed as shown in the example in figure 9-1. To retain the present value during a Modify (M) operation, you must <i>re-enter</i> the value displayed. Typing a carriage <i>return</i> with no value preceding it results in entry of a word filled with zeros; in a code segment, this results in a NOP instruction. After each command entry, PATCH prompts you for another command. To terminate PATCH, enter any character but <i>M</i> or <i>D</i> in response to a prompt; a carriage <i>return</i> is recommended for this purpose.

If an error occurs, PATCH calls the QUIT intrinsic to abort itself with one of the error numbers in table 9-2 appearing on your standard listing device.

```

:RUN PATCH.PUB.SYS

PROGRAM PATCHER CU.06
  FILE=?SLPATCH
?D,0,0,1
  140001
?D,0,0,5
  140001
  041001
  025026
  000600
  031020
?M,0,0,1
  140001,140002
?M,0,0,3
  140002,140001
  041001,041001
  025026,025026
? return

END OF PROGRAM

```

Figure 9-1. PATCH Example

Table 9-2. PATCH Error Numbers

ERROR NUMBER	MEANING
1	PATCH could not open the program file.
2	PATCH encountered an FREAD intrinsic error on the program file.
3	PATCH could not close the program file.
4	PATCH encountered an FREADDIR intrinsic failure in an operation other than Modify.
5	PATCH encountered an FREADDIR intrinsic failure in a Modify operation.
6	PATCH encountered an FWRITEDIR intrinsic failure in a Modify operation.

SLPATCH

SECTION

X

SLPATCH allows you to display or modify the contents of a segmented library (SL) file.

NOTE

You require no special capabilities to run this program. The program itself, however, runs in privileged mode because it must be able to access the system SL file even when that file is in use by other programs.

WARNING



This program can be used to modify the contents of privileged SLs, in which case the normal checks and limitations that apply to the standard users in MPE are bypassed. It is possible that modification of privileged SLs can destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from modification of privileged SLs. This service, which is not provided under the standard service contract, is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.

The system manager must ensure that users are not allowed access to any privileged SLs. He should restrict the use of SLPATCH only to necessary users, perhaps via a file lockword.

To use this program, follow the directions in table 10-1.

Table 10-1. Using SLPATCH

STEP	PROCEDURE
1	Log-on to the system. Then run SLPATCH by entering: <u>:RUN SLPATCH.PUB.SYS</u>
2	SLPATCH responds by printing a heading identifying itself, followed by a prompt for the name of the file you wish to modify or display: SL FILE PATCHER CU.06 SL FILE? Enter the name of the appropriate SL file. (The system SL file is named SL. PUB.SYS.)
3	SLPATCH prints a question mark (?) as a prompt for an SLPATCH command. ? To display the contents of an SL segment, enter: [segment-name,] D, segment-displacement [,number-of-words]

Table 10-1. Using SLPATCH (Continued)

STEP	PROCEDURE
3 (Continued)	To modify the contents of an SL segment, enter: <i>[segment-name,] M, segment-displacement [,number-of-words]</i> In these commands, all numbers and addresses are interpreted as <i>octal</i> values. The contents of the specified addresses are displayed as shown in the example in figure 10-1. If you omit the <i>number-of-words</i> parameter, one location is assumed. The <i>segment-name</i> parameter is needed only after the first prompt or when you wish to change segments. To enter a new value during a Modify (M) operation, type the new value after the value displayed as shown in figure 10-1. To retain the present value during a Modify (M) operation, enter an asterisk (*) in place of a new value. Typing a carriage-<i>return</i> with no value preceding it is prohibited; if you attempt this, you are prompted again to enter a value. After each command entry, SLPATCH prompts you for another command. To terminate SLPATCH, enter EXIT.

Fatal errors encountered when using SLPATCH may result in any of the following three error messages:

xx IO ERROR***

END OF FILE

FILE ERROR= *xx*

NOTE

In these messages, *xx* is an error code returned by the FCHECK intrinsic as described in the *MPE Intrinsics Reference Manual* (part no. 30000-90010).

SLPATCH then terminates, and MPE prints the following messages:

ABORT:SLPATCH.PUB.SYS.%0%660
PROGRAM ERROR #18:PROCESS QUIT. PARAM= 1

Non-fatal errors may result in one of the messages:

ILLEGAL COMMAND
SEGMENT NOT SPECIFIED
ILLEGAL SEGMENT NAME
INVALID SL FILE
ILLEGAL RANGE

```

:RUN SLPATCH.PUB.SYS

SL FILE PATCHER CU.06
SL FILE? SL.PUB.SYS
? SDMCOMM,D,0,5
025001
051404
041605
021040
005700
? D,0,5
025001
051404
041605
021040
005700
? M,0,1
025001, 025002
? M,0
025002, 025001
? EXIT

```

END OF PROGRAM

Figure 10-1. SLPATCH Example

SPOOK is a program that allows you to interrogate and operate on spooled devicefiles (spoolfiles) created and maintained by MPE.

NOTE

SPOOK runs under control of MPE, and only in on-line (session) mode.

No special MPE capabilities are required to run SPOOK, but certain program operations are restricted to users with Privileged Mode optional capability. In addition, a lack of certain capabilities (System Manager or Account Manager) may restrict the spoolfiles you can access.

WARNING

Do not transfer SPOOK to a pre-Series II HP 3000 Computer System — if you run this program on such a machine, it will destroy the system; in such a case, Hewlett-Packard cannot be responsible for system integrity.

SPOOK OPERATIONS

After initiating an interactive dialogue with SPOOK, you can request various operations by entering SPOOK commands. These functions and their related commands are:

Function	Command
• List characteristics of input/output spoolfiles.	SHOW
• Access (open) an output spoolfile to permit listing its contents (LIST command) or locating records within it (FIND command).	TEXT
• List the contents of all or part of an output spoolfile made available by the TEXT command.	LIST
• Locate a record by character string in an output spoolfile made available by the TEXT command, and list this record.	FIND
• Delete an output spool file.	PURGE

- | | |
|--|--------|
| • Modify the maximum width and request a control information display for lines listed by LIST and FIND commands. | MODE |
| • Change the output priority, number of copies requested, and/or destination device for an output spoolfile. | ALTER |
| • Terminate operation of SPOOK. | EXIT |
| • List all SPOOK commands and describe their syntax. | XPLAIN |
| • Enter the MPE Debug Facility. (Privileged-Mode users only.) | DEBUG |
| • Store output spoolfiles on magnetic tape. (System Manager/Supervisor users only.) | OUTPUT |
| • Restore output spoolfiles from magnetic tape to the system. (System Manager/Supervisor users only.) | INPUT |

Those commands that list output transmit it only to your terminal (the standard listing device) — you cannot specify any other output device.

For most of the above operations, you access a particular spoolfile by referencing either its *devicefileid* or *owner*. The *devicefileid* is a unique identifier consisting of a pound-sign (#) followed by the letter I (for input spoolfiles) or O (for output spoolfiles), followed by one to three digits (*nnn*). (The # and O are optional for output spoolfiles.) Thus, for input spoolfiles, the format is #*Innn*; for output spoolfiles, it is [#O]*nnn*. The *owner* is denoted by the name and account of the user who created the spoolfile, as reflected in the file label, in the format *username.acctname*.

SPOOK restricts access to spoolfiles on the basis of your log-on capability, as follows:

- System Manager or Supervisor: You may access any spoolfile in system.
- Account Manager: You may access any spoolfile in your log-on account.
- Standard User: You may access any spoolfile that you have created.

RUNNING SPOOK

To run SPOOK, log-on to the system and enter:

```
:RUN SPOOK.PUB.SYS
```

The program identifies itself by printing the following heading:

```
* SPOOK * (VERSION 4)
```

Then, the program prompts you for a command by displaying a greater-than sign (>). Respond by entering one of the commands discussed below.

ENTERING COMMANDS

Each command consists of a command name followed in most instances by a parameter list that is optional in some cases and required in others. The command name must begin immediately after the prompt character. Each command can contain up to 80 characters, and is terminated by a carriage-return. Blanks may appear anywhere between syntactic elements such as names, separators, or keywords, but not within a command name. Furthermore, a qualified identity such as *username*.*acctname* cannot contain embedded blanks. At your option, you may abbreviate command names and parameters as long as you avoid ambiguity. For instance, since each command name begins with a unique letter, you may abbreviate each such name by entering this letter alone. Examples appear in the discussion of each command.

At most terminals, you may request the following control functions with the control keys noted below:

Function	Key
Delete preceding character.	H ^c
Delete current line.	X ^c
Terminate current operation.	Y ^c

The SPOOK commands are described in the following pages. For easy reference, they are presented alphabetically by command name. For each, the following information appears:

- SYNTAX shows the command format. The symbols used in this format are defined under *CONVENTIONS USED IN THIS MANUAL* (Page viii).
- PARAMETERS describes the meaning, constraints, and default values of the command parameters.
- OPERATION discusses the function performed by the command.
- EXAMPLES shows the application of the command. The first example shows the command in its complete form; the succeeding examples show abbreviated forms.

ALTER

Changes output priority, number of copies requested, and/or destination device for output spoolfile.

SYNTAX

`ALTER devicefileid [;item [,item [,item]]]`

PARAMETERS

devicefileid Identifier of output spoolfile whose attributes are to be changed. For any such spoolfile, enter in format [#O] nnn. For current spoolfile accessed by TEXT command, enter an asterisk (*). (Required parameter.)

item Request to alter output priority, number of copies printed, or destination device of output spoolfile, in following format:

[PRI = *outputpriority*]

[COPIES = *numcopies*]

$$\left[\text{DEV} = \left\{ \begin{array}{l} \text{1dev} \\ \text{classname} \end{array} \right\} \right]$$

Each set of brackets, above, defines an item. You cannot use the same item more than once in the parameter list. The sub-parameters are:

outputpriority

New output priority for spoolfile. Used to select next spoolfile on disc for output, from among all those contending for a specific printer or card punch. Must be value ranging from 1 (lowest priority) to 13 (highest priority).

NOTE

When *outputpriority* is 1, output is always deferred. Thus, if you wish output written from disc, use *outputpriority* of 2 or greater;

numcopies

New value for number of copies of spoolfile to be listed. Must be value from 1 to 255.

1dev

Logical device number of new device on which the spoolfile is to be listed.

classname

Class name of new device on which spoolfile is to be listed. Must contain from 1 to 8 alphanumeric characters, begin with a letter, and terminate with any non-alphanumeric character such as a blank.

If you omit any *item* parameter, the characteristic denoted by that parameter remains unchanged. (Optional parameter.)

OPERATION

Changes output priority, number of copies requested, and/or destination device of output spoolfile.

ALTER

(Continued)

NOTE

This command can only be used with output spoolfiles in the READY, OPEN, or LOCKED state. You may use this command to alter the attributes of a file currently accessed by the TEXT command.

EXAMPLES

To change the currently-scheduled number of output copies to 3 for spoolfile #025, you may enter:

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

To change the output priority of a spoolfile currently accessed by the TEXT command to 7 and the destination device for this spoolfile to any device with the classname LP, enter:

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

NOTE

In the above example, both the command name and keywords are abbreviated.

DEBUG

Invokes MPE Debug Facility.

SYNTAX

DEBUG

PARAMETERS None

OPERATION

This command accesses the MPE Debug Facility and is used primarily by systems programmers. The Debug Facility allows programmers to establish break-points within programs and to display and modify data within the data stacks or registers used by those programs. The DEBUG command can be used to establish breakpoints within the SPOOK program and allows users to operate on SPOOK's data stack. The more general applications of the Debug Facility are discussed in *MPE Debug/Stack Dump Reference Manual*. Once you access this facility, you issue directives to it in response to interactive prompts (in the form of a question mark). These directives and various Debug messages are also discussed in the *MPE Debug/Stack Dump Reference Manual*.

NOTE

This command is available to you only if you have the Privileged Mode (PM) Optional Capability, described in *MPE Intrinsic Reference Manual*.

WARNING

When you run Debug in this way, you are operating in Privileged Mode and can easily destroy the system if you are not careful. Beware of modifying any locations or setting breakpoints in privileged code.

EXAMPLES

To call the Debug Facility, enter the following command. (In response, Debug displays a question mark to prompt you for a directive.)

```
>DEBUG
?
```

As with any SPOOK command, you may abbreviate the DEBUG command by entering simply:

```
>D
?
```


EXIT

Terminate operation of SPOOK.

SYNTAX

EXIT

PARAMETERS None

OPERATION Ends your dialogue with SPOOK, closes the currently-open text file, and terminates SPOOK.

EXAMPLES To exit from SPOOK, enter:

>EXIT

or

>E



FIND

Locates record in output spoolfile by character string.

SYNTAX

FIND [@] ['string']	[,count] recnumber1 * FIRST LAST [±offset] / recnumber2 * FIRST LAST [±offset] [ALL]
-------------------------	---

PARAMETERS @

Request to search entire record for *string*. If omitted, SPOOK only compares *string* against leading characters in record. (Optional parameter.)

string String by which SPOOK identifies record sought. May consist of any number of alphanumeric or special ASCII characters. Leading blanks in both *string* and record are removed by SPOOK. Parameter must be bounded by double apostrophes. If string is omitted, spoolfile is set to first record. (Optional parameter.)

NOTE

The next parameters in the parameter list specify the portion (range) of the spoolfile to be searched. If all of these are omitted, the current record (indicated by the current record-pointer setting) is sought.

recnumber1	Unsigned integer indicating position of first record (line) in range to be searched. To indicate first record in spoolfile, enter 0; to indicate second record, enter 1; and so forth. (Optional parameter.)
recnumber2	Unsigned integer indicating position of last record in range to be searched. Must be value greater than or equal to <i>recnumber1</i> . (Optional parameter.)
offset	Integer, preceded by plus or minus sign, that denotes an offset from record position indicated by immediately-preceding parameter. For instance, if preceding parameter is 32, then offset of +8 indicates record number 40 (the 41st record). (Optional parameter.)
*	Current record position (indicated by current record pointer). (Optional parameter.)
FIRST	First record in file, record 0. (Optional parameter.)
LAST	Last record in file (<i>recnumber</i> = (last record - 1)). (Optional parameter.)
count	Unsigned integer indicating number of records to be located beyond position denoted by preceding parameter. (Optional parameter.)
ALL	Request to locate all records in spoolfile. (Optional parameter.)

NOTE

If you indicate a beginning-record position but do not specify a last-record position, SPOOK locates all records from the specified beginning record position through the last record in the file.

FIND

(Continued)

OPERATION

FIND scans the spoolfile previously accessed by the TEXT command to locate the first record containing a character string that matches the *string* parameter. You can request SPOOK to scan the entire file or an inclusive portion within a specified range. In this scan, blanks and control characters at the beginning of each record and blanks at the beginning of string are ignored. If the record is found, it is displayed and the current record pointer is set to this record. If the record is not found, the current record pointer is set to one record beyond the last record in the range you specified (*recnumber* + 1) and no display occurs.

For each record located, the display shows:

- *Line number.* Record number showing record's position in file.
- *I/O controls.* Input/output controls applying to file, displayed only if requested by MODE command. In this display, W *nnn* indicates *Write with control nnn*; C *nnn* indicates *Control using nnn*; FOPEN indicates an *open file*; and FCLOSE indicates a *closed file*.
- *Record text.* Any leading control character is removed, and all other non-printing characters are each replaced by a period. If length of line displayed has been reduced to less than actual record length in file (by MODE command), record is truncated to the display length.

The display appears in the following format:

Line number	I/O Control	Record text
7	W	studies taken in spring of 1964, when Kaymeier

EXAMPLES

To search the entire spoolfile identified by the *devicefileid* #0250 for the record containing the string "NO. ERRORS," scanning only the leading characters in each record, enter:

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

To search this same file for the first record containing a colon (:), beginning with Record Number 21 and continuing through Record Number 30, scanning all characters in each record, enter:

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

To search this file for the first record containing an MPE :EOD command, beginning with the current record and continuing through the end-of-file, scanning only leading characters within each record, enter:

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

INPUT

Restores output spoolfile from magnetic tape to system.

SYNTAX

INPUT	[<i>username</i> [<i>.acctname</i>] <i>devicefileid</i> [<i>,devicefileid</i>] ...] ; <i>tapefile</i>
-------	--

PARAMETERS

<i>username</i>	User name under which spoolfiles to be restored were created, or @ to denote files for all users. (Optional parameter.)
<i>acctname</i>	Account name under which spoolfiles to be loaded were created, or @ to denote spoolfiles for all accounts. (Optional parameter.)
<i>devicefileid</i>	Identifier of output spoolfile to be restored, in format [#O] <i>nnn</i> . (Required parameter if requesting individual spoolfiles.)

NOTE

When *username* [*.acctname*] and *devicefileid* are both omitted, all files belonging to the log-on *username.acctname* are restored.

<i>tapefile</i>	Name of magnetic tape file on which files to be retrieved now reside. This file must be referenced in back-reference (*) format, which refers to previous MPE :FILE command that defines this file as residing on magnetic tape. Message is output to console operator requesting him to mount tape identified by <i>filereference</i> parameter in :FILE command, and allocate tape unit to you. (Required parameter.)
-----------------	---

OPERATION

Reads back into system (on disc) one or more output spoolfiles previously stored offline (on magnetic tape) by the SPOOK command OUTPUT.

NOTE

You must have System Manager or System Supervisor capability to use the INPUT command.

As long as the files exist on the tape, you can request them individually by *devicefileid* or collectively by *username.acctname*. (In either case, however, the user and account names under which the spoolfiles were created must still be defined in the system.) SPOOK reads the spoolfiles from a single reel of tape or from multi-reel tape files. (For a discussion of system/operator interaction for multi-reel file handling, see the description of the OUTPUT command.) As each spoolfile is restored, it is changed to reflect:

- A new *devicefileid*, for future reference.
- A change to the session/job number under which the spoolfile was created, to include an apostrophe as follows:

For sessions: #S '*nnn*

For jobs: #J '*nnn*

When subsequent MPE or SPOOK commands display the session/job number, the apostrophe indicates that the spoolfile was copied into the system from an external source and may have originated on a system with a different configuration.

- A possible change to the device class name and logical device number, if the spoolfile was restored to a different disc device.

INPUT

(Continued)

For each spoolfile restored, SPOOK indicates the above changes by displaying the following information:

<i>Heading</i>	<i>Item</i>
#FILE (First Column)	Original <i>devicefileid</i> still assigned.
#FILE (Second Column)	New <i>devicefileid</i> assigned.
# JOB	New session/job number assigned.
DEV/CL	New logical device number/class name assigned.
OWNER	The <i>username.acctname</i> of spoolfile creator.

The output display appears in this format:

#FILE	==>	#FILE	#JOB	DEV/CL	OWNER
#099	==>	#0105	#S'44	LP	ALAN.TP
#0100	==>	#0106	#S'44	LP	ALAN.TP
↑		↑	↑	↑	↑
Old		New	New	New	Creator's
<i>devicefileid</i>		<i>devicefileid</i>	session/ job number	class name	name and account

For any spoolfiles that are not restored according to your request, SPOOK displays an error message.

NOTE

Before issuing an INPUT command, you must identify *tapefile* as a magnetic tape file. Do this with the MPE :FILE command, in the following format, including only those parameters shown:

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

The *device* parameter must indicate the device class name or logical unit number of a magnetic tape unit. All other parameters for *tapefile* are supplied by the INPUT command executor; if you attempt to supply any of these, SPOOK rejects your INPUT command.

EXAMPLES

To restore all spoolfiles from a tapefile named T, enter:

```
:FILE T; DEV=TAPE
:RUN SPOOK.PUB.SYS
.
.
.
>INPUT @.@; *T
```

To restore the spoolfiles identified by the *devicefileids* #03, #04, and #07 from this same *tapefile* (with the above :FILE command also in effect in this case), enter:

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

To restore all spoolfiles belonging to the user and account named U.A (again with the above :FILE command in effect), enter:

```
>IN U.A; *T
```

LIST

Lists contents of spoolfile.

SYNTAX

LIST	[,count]	
	$\left[\begin{array}{c} \text{recnumber1} \\ * \\ \text{FIRST} \\ \text{LAST} \end{array} \right] [\pm\text{offset}]$	$/ \left[\begin{array}{c} \text{recnumber2} \\ * \\ \text{FIRST} \\ \text{LAST} \end{array} \right] [\pm\text{offset}]$
	[ALL]	

PARAMETERS

The parameter list specifies the portion (range) of the spoolfile to be listed. If all parameters are omitted, the current record (indicated by the current record-pointer setting) is displayed.

<i>recnumber1</i>	Unsigned integer indicating position of first record in range to be listed. To indicate first record in spoolfile, enter 0; to indicate second record, enter 1; and so forth. (Optional parameter.)
<i>recnumber2</i>	Unsigned integer indicating position of last record in range to be listed. Must be value greater than or equal to <i>recnumber1</i> . (Optional parameter.)
<i>offset</i>	Integer, preceded by plus or minus sign, that denotes an offset from record position indicated by immediately-preceding parameter. For instance, if preceding parameter is 50, then an offset of +7 indicates record number 57 (the 58th record). (Optional parameter.)
*	Current record position (indicated by current-record pointer). (Optional parameter.)
FIRST	First record in file, record 0. (Optional parameter.)
LAST	Last record in file (<i>recnumber</i> = (last record - 1)). (Optional parameter.)
<i>count</i>	Unsigned integer indicating number of records to be listed beyond position denoted by preceding parameter. (Optional parameter.)
ALL	Request to list all records in spoolfile. (Optional parameter.)

NOTE

If you indicate a beginning record position, but do not specify a last-record position, SPOOK displays only the record indicated by the beginning-record position.

OPERATION

LIST displays at your terminal the contents of the spoolfile accessed by the TEXT command. You may request the entire file contents, or an inclusive portion within a specific range. For each record (line), the display presents the following items:

- *Line number.* Record (line) number of record, showing its position in file.
- *I/O controls.* Input/output controls applying to file, displayed only if requested by MODE. In this display, W *nnn* indicates *Write with control nnn*; C *nnn* indicates *Control using nnn*; FOPEN indicates an *open file*; and FCLOSE indicates a *closed file*.
- *Record text.* Any leading control character is removed, and all other non-printing characters are each replaced by a period. If the length of the displayed lines has been reduced to less than actual record length (by MODE), record listed is truncated to display length.

The display appears in the following format:

LIST

(Continued)

Line number	I/O Control	Record text
0	FOPEN	
1	W	PAGE 1 HEWLETT-PACKARD 32201A.5.02 EDIT/3000 SAT. DEC 11, 1976
2	W	
3	W	
4	W	The data produced by the Program Taurus appears
5	W	below. It is extrapolated on the basis of a

When the listing is complete, the current record pointer is set to the record immediately following the last record position specified in the LIST command.

EXAMPLES

To display all records in the spoolfile identified by the *devicefileid* #016, enter

```
>TEXT #016  
>LIST ALL
```

To display the current record on this same text file, enter:

```
>L
```

To display the range beginning with the second record after the current record, and ending with the fourth record before the last, enter:

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

To display the range beginning with record number 5 and continuing for four records beyond, enter:

```
>L 5, 4
```

NOTE

This entry actually lists records number 5 through 8.

MODE

Modifies maximum line width and enables control information display for records listed by FIND and LIST commands.

SYNTAX

MODE	$\left[\left\{ \begin{array}{l} \text{WIDTH} = \left\{ \begin{array}{l} \pm nnn \\ \text{OFF} \end{array} \right\} \\ \text{CONTROLS} = \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} \end{array} \right\} \left[, \left\{ \begin{array}{l} \text{WIDTH} = \left\{ \begin{array}{l} \pm nnn \\ \text{OFF} \end{array} \right\} \\ \text{CONTROLS} = \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} \end{array} \right\} \dots \right]$
------	---

PARAMETERS The parameter list changes the maximum width of the lines displayed by the FIND and LIST commands, and enables display of input/output controls found in the spoolfile by these commands.

<i>nnn</i>	Integer preceded by plus or minus sign, indicating maximum line width for FIND or LIST output; plus prefix denotes width in words, and minus prefix specifies width in bytes. (Optional parameter.)
OFF	When used with WIDTH keyword, specifies that actual record width is used for output display lines. When used with CONTROLS keyword, disables display of input/output control information in output lines. (Optional parameter.)
ON	Enables display of input/output control information in output lines. (Optional parameter.)

NOTE

When any parameter is omitted from the MODE command, SPOOK assumes that the previous value is in effect.

OPERATION Modifies maximum line width and enables control information display for records listed by subsequent FIND and LIST commands.

NOTE

When you initially access SPOOK, this program assumes that the OFF option is in effect for both line width and control display.

EXAMPLES To set the maximum line width to 60 characters (bytes), and enable the mode-control display, enter:

>MOD WIDTH=-60, CONTROLS=ON

To restore the display line width equal to the actual record size, enter:

>M W=OFF

The MODE controls established in the first example remain set until disabled by:

>M C=OFF



OUTPUT

Stores output spoolfiles onto magnetic tape.

SYNTAX

```
OUTPUT [ username[.acctname]  
        devicefileid[,devicefileid]... ] ;tapefile
```

PARAMETERS	<i>username</i>	User name under which spoolfiles to be stored were created, or @ to store files for all users. (Optional parameter.)
	<i>acctname</i>	Account name under which spoolfiles to be stored were created, or @ to store files for all accounts. (Optional parameter.)
	<i>devicefileid</i>	Identifier of output spoolfile to be stored, entered in format [#O] <i>nnn</i> . (Required parameter if requesting individual spoolfiles.)

NOTE

When *username* [*.acctname*] and *devicefileid* are both omitted, all files belonging to the log-on *username.acctname* are stored.

<i>tapefile</i>	Name of magnetic tape file on which spoolfiles are to be stored. This file must be specified in the <i>back-reference</i> (*) format, which refers to a previous MPE :FILE command that defines this file as residing on magnetic tape. A message is output to the console operator requesting him to mount the tape identified by the <i>filereference</i> parameter in the :FILE command, and allocate the tape unit to you. (Required parameter.)
-----------------	--

OPERATION

Copies onto magnetic tape one or more output spoolfiles. These spoolfiles may be restored to the system later with the INPUT command.

NOTE

You must have System Manager or System Supervisor Capability to use the OUTPUT command.

You may store only output spoolfiles that exist in the READY or LOCKED state. You may store them on a single reel of tape, or on multi-reel tape files. If you do the latter, SPOOK requests the console operator to change reels at the appropriate times by transmitting the following message to the console:

CHANGE REELS ON LDEV nn? (Y/N)

To change reels, the operator enters *Y* and mounts the new reel. To abort this operation, he enters *N*.

If the operator mounts the wrong reel, SPOOK transmits this message to the console:

INCORRECT REEL — TRY AGAIN? (Y/N)

As each spoolfile is copied, SPOOK changes its output priority to 1 (deferring its output from disc) but retains the spoolfile in the system. If you subsequently raise the output priority (via the ALTER command), the file may be copied from disc at that time.

OUTPUT

(Continued)

For each spoolfile stored, SPOOK lists the following information:

<i>Heading</i>	<i>Item</i>
#FILE	Present <i>devicefileid</i> .
#JOB	Session/job number of creating session/job.
DEV/CL	Logical device number/class name assigned.
SECTORS	Total number of disc sectors occupied by spoolfile.
OWNER	The <i>username.acctname</i> of spoolfile creator.

The output display appears in this format:

#FILE	#JOB	DEV/CL	SECTORS	OWNER
#099	#S44	LP	8	ALAN.TP
#0100	#S44	LP	12	ALAN.TP

For any spoolfiles not copied to tape according to your request, SPOOK displays an error message.

NOTE

Before issuing an OUTPUT command, you must identify *tapefile* as a magnetic tape file. Do this by using the MPE :FILE command, written in the following format, including only those parameters shown:

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

The *device* parameter must indicate the device class name or logical device number of a magnetic tape unit. All other parameters are supplied by the OUTPUT command executor; if you attempt to supply any of these, SPOOK rejects the OUTPUT command.

EXAMPLES

To store all output spoolfiles onto a tapefile named T, enter:

```
:FILE T; DEV=TAPE  
:RUN SPOOK.PUB.SYS
```

```
>OUTPUT @.@; *T
```

To store all output spoolfiles created under the user and account named U.A (with the above :FILE command in effect), enter:

```
>O U.A; *T
```

To store the output spoolfiles having the *devicefileids* #012 and #01, (again with the above :FILE command in effect), you may enter:

```
>OUT #012, 4; *T
```

PURGE

Deletes output spoolfiles.

SYNTAX

PURGE *devicefileid* [*,devicefileid*] ...

PARAMETERS *devicefileid* Spoolfile identifier, entered in format [#O] *nnn* (for any output spoolfile) or * (for spoolfile currently accessed by TEXT command). (Required parameter.)

OPERATION Deletes one or more output spoolfiles. You may request this operation for output spoolfiles in the READY or LOCKED state only. Also, you may request deletion of a spoolfile currently accessed by the TEXT command.

For each spoolfile deleted, SPOOK lists the following information:

<i>Heading</i>	<i>Item</i>
#FILE	<i>Devicefileid</i> assigned.
#JOB	Session or job number of the session or job that created the file.
DEV/CL	Logical device number or device class name assigned.
SECTORS	Total number of sectors occupied by spoolfile.
OWNER	The <i>username.acctname</i> of spoolfile creator.

The output display has the following format:

#FILE	#JOB	DEV/CL	SECTORS	OWNERS
#0367	#S156	LP	12	MAC.TECHPUBS
#0368	#S156	LP	12	MAC.TECHPUBS
#0369	#S156	LP	12	MAC.TECHPUBS

For any spoolfiles not purged according to your request, SPOOK displays an error message.

EXAMPLES

To delete the spoolfile with the *devicefileid* #024, enter:

>PURGE #024

To delete the spoolfiles with the *devicefileids* #019, #020, and #021, you may enter:

>P 19, 20, 21

SHOW

Lists characteristics of input/output spoolfiles.

SYNTAX

```
SHOW [ [username [acctname ] ] [ ; [ @ ] [ I ] [ O ] ] ]  
      devicefileid [ ,devicefileid ] ...
```

PARAMETERS

username User name under which spoolfile to be listed was created, or the character @ to denote spoolfiles for all users. (Optional parameter.)

acctname Account name under which file to be listed was created, or @ to denote spoolfiles for all accounts. (Optional parameter.)

NOTE

When you specify @.ACCOUNT1, all spoolfiles for ACCOUNT1 are listed. When you specify JONES.@, all spoolfiles identified by the user name JONES under all accounts are listed. When you specify @.@, all spoolfiles in the system are listed. When you omit *username* and/or *acctname*, all spoolfiles under your current (log-on) user name and account name are listed.

To request spoolfiles under all accounts, (*username.@* or *@.@*), you must have System Manager or Supervisor capability.

To request all files under any account (*@.acctname*), you must have Account Manager capability for that account.

@ Request to list output in long format. If omitted, short format applies. Both formats are illustrated under **OPERATION** below. (Optional parameter.)

I Request to display input spoolfiles only. (Optional parameter.)

O Request to display output spoolfiles only. (Optional parameter.)

NOTE

If you specify neither I nor O, both input and output spoolfiles are displayed.

devicefileid Spoolfile identifier. To specify an input spoolfile, enter in format #Innn. To specify an output spoolfile, enter in format [#O]nnn. To specify spoolfile currently accessed by TEXT command, enter an asterisk (*). When you use the *devicefileid* parameter, long-format output always appears. (Optional parameter.)

OPERATION

Displays characteristics of selected spoolfiles, in short or long format, on your standard listing device. In short format, the following information appears:

- *Devicefileid.*
- *Session/job number*, identifying session or job in which spoolfile was created.
- *File name.*
- *File state:*

SHOW

(Continued)

ACTIVE For input spoolfile, input is being read from spooled device to disc. For output spoolfile, spoolfile on disc is being written to printer, plotter, or card punch.

READY For input spoolfile, spooling is complete and file is now ready for use by a program. For an output spoolfile, spoolfile is stored on disc and ready for output.

OPENED For input or output spoolfile, spoolfile is being accessed by a program.

LOCKED For output spoolfile only, READY but system is using spoolfile with exclusive access.

- *Owner* (username.acctname of file creator).

The short format appears as follows:

#FILE	#JOB	FNAME	STATE	OWNER
#0367	#S156	LISTER	READY	MAC.TECHPUBS
#0368	#S156	LISTER	READY	MAC.TECHPUBS
#0369	#S156	LISTER	READY	MAC.TECHPUBS

In the long format, the following items may appear in addition to the short-format information:

- *Logical device number* or device class *name* of originating device (for input spoolfiles) or destination device (for output spoolfiles).
- *Output priority* of file.
- *Number of copies* requested (for output spoolfiles only).
- *R*, indicating a re-startable spooled job file.
- *F*, indicating that a forms-alignment message applies to the spoolfile.
- *N*, indicating that insufficient disc space was available when spoolfile was created.
- *Logical device number* of device where file label is stored.
- *Sector address* of spoolfile label.
- *Total number of disc sectors* used by spoolfile.
- *Total number of lines (records)* in spoolfile.
- *Time* spoolfile entered READY state.

The long format appears as follows:

#FILE	#JOB	FNAME	STATE	DEV/CL	PR	COP	RFN	OWNER
#0367	#S156	LISTER	READY	LP	1	1		MAC.TECHPUBS
#0368	#S156	LISTER	READY	LP	1	1		MAC.TECHPUBS
#0369	#S156	LISTER	READY	LP	1	1		MAC.TECHPUBS
#FILE	LDEV	LABEL	SECTORS		LINES		TIME	
#0367	%2	%205277	12		34		12:55	12/11/76
#0368	%3	%130315	12		30		12:59	12/11/76
#0369	%4	%200360	12		34		13: 1	12/11/76

For each spoolfile requested that cannot be displayed, an error message appears.

SHOW

(Continued)

EXAMPLES

To display, in long format, the characteristics of all spoolfiles created under the user name SMITH and account name ALPHA, enter:

>SHOW SMITH.ALPHA; @

To display, in short format, the attributes of all spoolfiles created under your log-on user name and account, enter simply:

>S

To display, in short format, the characteristics of all output spoolfiles, enter:

>S @.@; 0

To display, in long format, the characteristics of the spoolfiles identified by the *devicefileids* #I12, #O32, and #O23, enter:

>S #I12, #O32, 23

TEXT

Opens output spoolfile for access by
FIND and LIST commands.

SYNTAX

TEXT <i>devicefileid</i>

PARAMETERS

devicefileid

Spoolfile identifier, entered in format [#O] *nnn* (for any output spoolfile)
or * (for spoolfile currently accessed by TEXT command.)

OPERATION

Opens an output spoolfile in READY state, transferring it to LOCKED state to permit you to access it with the FIND and LIST commands. If you enter a TEXT command for a spoolfile already opened by a previous TEXT command, SPOOK closes that spoolfile and returns it to the READY state. When SPOOK terminates, it automatically closes any spoolfile opened through TEXT.

EXAMPLES

To open the spoolfile identified as #022 and place it in the LOCKED state, enter:

>TEXT #022

or

>T #022

or

>T 22

XPLAIN

List, and describe syntax of, all
SPOOK commands.

SYNTAX

XPLAIN

PARAMETERS None

OPERATION Displays all commands available through SPOOK and presents an abbreviated description of their syntax. The display appears in the following format:

```
DEBUG
EXIT
XPLAIN
SHOW  [ USER [ .ACCOUNT ] ] [ ; [ @ ] [ I ] [ O ] ]
SHOW  DEVICEFILEID [ , DEVICEFILEID ]....
TEXT  DEVICEFILEID
LIST  [ RANGE ]
FIND  [ @ ] [ 'STRING' ] [ , FRANGE ]
MODE  [ OPTION [ , OPTION ]...]
      OPTION = WIDTH / CONTROLS
ALTER DEVICEFILEID [ ; OPTION [ , OPTION ]....]
      OPTION = PRI / COPIES / DEV
PURGE DEVICEFILEID [ , DEVICEFILEID ]....
INPUT [ USER [ .ACCOUNT ] ] ; TAPEFILE
INPUT DEVICE
OUTPUT [ USER [ .ACCOUNT ] ] ; TAPEFILE
OUTPUT DEVICEFILEID [ , DEVICEFILEID ].. ; TAPEFILE
```

EXAMPLE

To display the SPOOK commands and their syntax, enter:

>XPLAIN

or

>X

MESSAGES

During your interaction with SPOOK, two types of messages may appear: warning messages and error messages. Both types are summarized in Table 11-1, with warning messages denoted as Type W and error messages as Type E.

Warning messages typically indicate an inability to initiate a requested command, or truncation of some of the output from an executed command. They appear in this format:

***WARNING** = *msgno* * . . . *message*

In this format, *msgno* is a two-digit code for the message and *message* is the message text.

Error messages may occur immediately after command entry, when a command syntax error or an illegal parameter is detected. They may also appear following displays from SHOW, PURGE, INPUT, or OUTPUT operations to indicate that the requested operation did not take place for a particular spoolfile. They appear in this format:

[#FILE] ***ERROR** = *msgno* [BYTE = *byteno*] * . . . *message*

The notation #FILE appears only in messages pertaining to output displays; *msgno* is the two-digit code for the message; *byteno* identifies the erroneous character causing a syntax or parameter error; *message* denotes the message text.

Table 11-1. SPOOK Messages

Msg No.	Type	Message	Meaning/Action
1	W	NOT INTERACTIVE SESSION	You tried to initiate SPOOK during batch job. Log-on in session mode, initiate SPOOK again.
2	W	END OF FILE	SPOOK encountered end-of-file indicator within your response to prompt for input. SPOOK automatically terminates; run SPOOK again and respond to prompt correctly.
3	W	TOO MANY FILES	You have referenced too many files in your last command for SPOOK's file directory to handle. For instance, you may have requested SHOW for all files when very many exist in system. Re-issue command for fewer files.
4	W	INSUFFICIENT CAPABILITY	You requested operation that demands an MPE capability that you do not possess, such as System Manager, Account Manager, or Privileged Mode. Re-enter command with less restrictive parameters or select new command.
19	E	IMPOSSIBLE INTERNAL ERROR	System error internal to SPOOK occurred. Call your HP Customer Engineer.
20	E	INVALID COMMAND NAME	You entered improper command name. Re-enter command with proper name.

Table 11-1. SPOOK Messages (Continued)

Msg No.	Type	Message	Meaning/Action
21	E	COMMAND NAME TOO BIG	You entered command name that exceeded six characters. Re-enter command with proper name.
22	E	PROMPT I/O ERROR	Input/output system transmission error occurred while SPOOK was writing prompt to your terminal. SPOOK attempts to write prompt again.
23	E	INPUT I/O ERROR	Input/output system transmission error occurred while SPOOK was reading input from your terminal. SPOOK again attempts to read INPUT.
24	E	UNABLE TO CLOSE FILE	SPOOK cannot close spoolfile, as when it fails to return previous TEXT file to system during execution of new TEXT command. Re-enter last command again.
25	E	UNABLE TO PURGE FILE	SPOOK cannot delete spoolfile you referenced in PURGE command. Re-enter PURGE command again.
26	E	FILE READ ERROR	SPOOK cannot read spoolfile currently accessed by TEXT command during FIND or LIST operation. Request FIND or LIST operation again; if this fails, request TEXT operation for this file again and then request FIND or LIST operation.
27	E	FILE FCONTROL ERROR	SPOOK cannot perform an input/output control operation, such as re-winding tape. Request operation again.
28	E	FILE NOT 'READY'	You attempted to access spoolfile that is not in READY state; for example, perhaps you tried TEXT operation on an ACTIVE file. Select another file or wait until state of desired file changes.
29	E	UNABLE TO OPEN FILE	SPOOK cannot open requested spoolfile in TEXT or PURGE command. Select another file or wait until this spoolfile enters READY state.
30	E	INPUT FILE NOT ALLOWED	You cannot use this command to access input spoolfile. Select output spoolfile or enter different command.
31	E	FILE NOT FOUND	SPOOK cannot locate spoolfile you requested. Check to determine if you referenced correct file name or <i>devicefileid</i> , and re-enter command.
32	E	INVALID FILE ID	You referenced an incorrect <i>devicefileid</i> . Check with SHOW command to ensure that this <i>devicefileid</i> exists on system, and re-enter command.
33	E	UNEXPECTED CHARACTER	Your last command included syntax error. Check proper syntax and re-enter command.
34	E	USER NAME TOO BIG	You referenced user name longer than eight characters. Re-enter correct name.
35	E	USER NOT ACCESSIBLE	You tried to access spoolfile in improper user domain, or referenced user name not defined in system. Select proper name and re-enter command.

Table 11-1. SPOOK Messages (Continued)

Msg No.	Type	Message	Meaning/Action
36	E	ACCOUNT NAME TOO BIG	Referenced account name was longer than eight characters. Re-enter correct name.
37	E	ACCOUNT NOT ACCESSIBLE	You tried to access spoolfile in improper account, or referenced account name not defined in system. Select proper name and re-enter command.
38	E	INVALID LINE MNEMONIC	You did not enter FIRST or LAST parameter properly in the FIND or LIST command. Check command syntax and re-enter it properly.
39	E	INVALID LINE NUMBER	Referenced record number was out-of-range or illegal (perhaps containing an alphabetic character) in FIND or LIST command. Re-enter command properly.
40	E	INVALID LINE COUNT	You used invalid <i>count</i> parameter in FIND or LIST command. Check syntax and re-enter command properly.
41	E	INVALID LINE RANGE	Beginning-record parameter value exceeded last-record parameter value in FIND or LIST command. Re-enter command with correct parameter values.
42	E	NON TERMINATED CHARACTER STRING	You omitted at least one of the quotation marks surrounding <i>string</i> parameter in FIND command. Re-enter command with quotation marks around string.
43	E	INVALID OPTION NAME	You used an improper keyword in ALTER or MODE command. Check command syntax and re-enter command properly.
44	E	INVALID OPTION SEPARATOR	You omitted equal sign following keyword in ALTER or MODE command. Re-enter command properly.
45	E	INVALID OPTION PARAMETER	Improper parameter followed keyword in ALTER or MODE command. Check command syntax and re-enter command properly.
46	E	NO TEXT FILE	FIND or LIST operation attempted for spoolfile not opened by TEXT command. Enter TEXT command for this spoolfile, then enter FIND or LIST.
47	E	FILE NOT 'READY/OPEN'	ALTER operation requested for spoolfile not in READY or OPEN state. Select another spoolfile or command — it is too late to access this spoolfile.
48	E	TEXT FILE NOT ALLOWED	You tried to access current TEXT file through command that cannot request this file. Select another command or spoolfile.
49	E	MISSING SEMI COLON	You used another character in place of required semicolon in INPUT or OUTPUT command. Re-enter command correctly.
50	E	UNABLE TO OPEN TAPE FILE	SPOOK cannot open requested tape file during INPUT or OUTPUT command execution. Check to see if tape unit is on-line or if tape is properly mounted.

Table 11-1. SPOOK Messages (Continued)

Msg No.	Type	Message	Meaning/Action
51	E	UNABLE TO CLOSE TAPE FILE	SPOOK cannot close requested tape file during INPUT or OUTPUT command execution. Unload tape file manually.
52	E	INVALID TAPE FILE	You entered improper file designator in INPUT or OUTPUT command. Check preceding FILE command, and be sure you have used proper back-reference format in all commands.
53	E	INVALID TAPE FORMAT	In an INPUT command, you referenced a tape file not previously stored through OUTPUT command. Select new tape or command.
54	E	TAPE FILE READ ERROR	SPOOK cannot read tape file during INPUT or OUTPUT command operation. Re-enter command.
55	E	TAPE FILE WRITE ERROR	SPOOK cannot write to tape file during INPUT or OUTPUT command operation. Re-enter command.
56	E	USER.ACCOUNT NOT ALLOWED	You tried to reference spoolfile by its <i>username.acctname</i> in PURGE command, but this type of reference is not allowed. Re-enter command with proper specific file reference.
57	E	NO EQUIVALENT DEVICE	During INPUT operation, SPOOK encountered file for which no equivalent appropriate device exists on system, perhaps because spoolfile was previously output on different system. See your computer operator.
58	E	NO EQUIVALENT CLASS	During INPUT operation, SPOOK encountered file for which no appropriate equivalent device class name is defined in system, perhaps because spoolfile was previously output on different system. See your computer operator.
59	E	NO ROOM IN DEVICE TABLE	SPOOK cannot restore all spoolfiles requested in INPUT command because spoolfile directory in system is full. Re-enter INPUT command later.
60	E	MULTI REEL ABORT	You aborted a multi-reel INPUT or OUTPUT operation by entering N in response to the prompt CHANGE REELS ON LDEV nn? (Y/N) If this action was a mistake, re-enter INPUT or OUTPUT command to begin operation again.

MEMLOGAN allows you to display the history of memory errors as recorded by the Memory Error Logging System.

NOTE

No special MPE user capabilities are required to run this program. However, you can run it from a group/account other than PUB.SYS only if your System Manager has released the security provisions of the program file MEMLOGAN.PUB.SYS (through the MPE :RELEASE command) or if your log-on account is assigned System Manager Capability.



MEMORY-LOGGING OPERATIONS

Memory error logging is performed by the system process MEMLOGP. MPE always initiates MEMLOGP during system initialization if error-correcting memory is present on the system. Once initiated, MEMLOGP periodically obtains the latest error information from the error-correcting memory logging boards and records it on the disc file MEMLOG.PUB.SYS, continuing as long as the system remains in operation. By system default, MEMLOGP checks for errors immediately after it is initiated and every 60 minutes thereafter. (You can alter this 60-minute interval by running the program MEMENTIMER, discussed in Section XIII.) When MEMLOGP detects errors that occurred since its last interrogation, MEMLOGP updates the two-record MEMLOG file to reflect these errors. (For further information about error-logging, please see *HP 3000 Series II Console Operator's Guide*.)

NOTE

The Memory Logging Facility is NOT related in any way to the System Logging Facility that records events on the System Log File; the operation of either of these facilities has no effect at all upon the operation of the other.

RUNNING MEMLOGAN

The program MEMLOGAN reads the error history recorded in MEMLOG.PUB.SYS and writes it to an output device as a formatted display. (The MEMLOG file must be released by the System Manager (with the :RELEASE command) before MEMLOGAN can be successfully run under a group and account other than PUB.SYS.) To run MEMLOGAN, log-on to the system and enter:

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

In this command, the subparameter *n* is a one-digit code that requests the options indicated below:

<i>n</i>	<i>Option</i>
0	MEMLOGAN displays the current contents of MEMLOG but makes no changes to this file. (If you omit the <i>n</i> parameter from the :RUN command, this option occurs by default.)
1	MEMLOGAN displays the current contents of MEMLOG, deletes all previously-recorded errors, and then sets this file to the NO-ERROR state.

NOTE

Execute MEMLOGAN with this option immediately after bringing up a system with error-correcting memory for the first time or after changing the size of memory beyond 128K words. This ensures a clear MEMLOG file and valid subsequent error counts.

2	MEMLOGAN displays the current contents of MEMLOG and then deletes this file from the system. (Because of the security provisions assigned to this file, this is the only way to purge this file from the system.)
---	---

NOTE

As a general rule, only the System Manager selects this option.

By default, MEMLOGAN writes all output to your standard listing device (\$STDLIST). You may, however, re-direct the output to another file by using a :FILE command referencing the formal file designator OUT:

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

MEMLOGAN OUTPUT

The output format varies according to whether MEMLOG is clear and in the NO-ERROR state, has been updated with no errors, or has been updated to reflect errors, as described below.

If MEMLOG is clear and in the NO-ERROR state (as it would be after MEMLOGAN was executed with the PARM=1 option), MEMLOGAN displays the following message and terminates:

*** NO ENTRIES IN MEMLOG FILE ***

If MEMLOG has been updated with no errors, MEMLOG prints the date and time of the first and last log updates, and the logging interval at the time of the last log updates. The interval appears in the following format:

hh:mm:ss

In this format, *hh* indicates hours, *mm* represents minutes, and *ss* denotes seconds. MEMLOGAN then displays the following message and terminates:

*** NO ERRORS LOGGED ***

If MEMLOG has been updated with errors logged, MEMLOGAN prints the date and time of the first and last log updates, and the logging interval at the time of the last log update, in the format noted above. MEMLOGAN also displays the date and time of the first and last errors logged, followed by a tabular representation of the information in MEMLOG that appears in the format shown below:

I	ADDRESS			I	ERROR TYPE				I	ERROR I

I	MEMORY	I	LOC	I	ROW	I	TYPE	BIT	BOARD	CHIP I COUNT I

I		I		I		I				I
I		I		I		I				I
I	mem	I	loc	I	row	I	type	bit	board	chip I cnt I
I		I		I		I				I
I		I		I		I				I

In this table, the variables shown above have the following meanings:

<i>mem</i>	The memory bank where the error occurred, shown as a digit from 0 through 3. Banks 0 and 1 refer to the lower 128K words of memory; Banks 2 and 3 refer to the upper 128K words.
<i>loc</i>	The lower or upper 32K words within the bank indicated by <i>mem</i> , shown as L32K or U32K respectively.
<i>row</i>	The row designation on the board in which the failing chip is located, indicated as a digit from 0 through 7.
<i>type</i>	Type of error detected, as follows:
CHECK	Check bit error.
DATA	Data bit error.
LOGGING SYSTEM ERROR	Error-logging hardware failure. Represents error on error-logging input/output card rather than true memory error.
MULTIPLE BIT ERROR	Error in more than one bit.
I/O DRIVER FAULT	Check bit input/output driver fault.
READ CLOCK FAILURE	Failure in reading system clock.

bit If *type* = CHECK , *bit* refers to the failing check bit—A, B, C, or D.
 If *type* = DATA, *bit* refers to the failing data bit—0 through 15.

board Board on which error occurred. If *type* = CHECK, board may indicate:

SMA Semi-conductor memory array board. Check Bit A only.
 FCA Fault-correcting array board. Check Bits B, C, D, or E.

If *type* = DATA, *board* may indicate only:

SMA Semi-conductor memory board.

chip Chip on which error occurred, in format:

Un

The variable *n* is a digit indicating the chip number.

cnt The number of logging intervals during which this error was detected at least once.

NOTE

This value does NOT represent the number of times that an error was actually detected.

An example of MEMLOGAN output appears in figure 12-1.

LOGGING STARTED	-	DATE: 10/ 6/76	TIME: 15:30
FIRST ERROR LOGGED	-	DATE: 10/ 6/76	TIME: 17:26
LAST ERROR LOGGED	-	DATE: 10/14/76	TIME: 10:00
LAST LOG UPDATE	-	DATE: 10/14/76	TIME: 16:20
TIMING INTERVAL	-	1:00:00	

I	ADDRESS			I	ERROR TYPE				I ERROR I		

I	MEMORY	I	LOC	I	ROW	I	TYPE	BIT	BOARD	CHIP	I COUNT I

I	BANK 1	I	L32K	I	0	I	DATA	10	SMA	U913	I 9 I
I		I		I	7	I	DATA	9	SMA	U112	I 39 I
I	BANK 2	I	L32K	I	4	I	CHECK	D	FCA	U45	I 2 I
I		I		I	5	I	CHECK	C	FCA	U34	I 1 I
I		I		I	5	I	DATA	5	SMA	U38	I 1 I
I		I		I	7	I	CHECK	C	FCA	U14	I 1 I
I		I	U32K	I	6	I	CHECK	B	FCA	U27	I 2 I
I		I		I	7	I	CHECK	E	FCA	U110	I 4 I
I	BANK 3	I	L32K	I	0	I	DATA	14	SMA	U917	I 1 I
I		I		I	3	I	DATA	9	SMA	U612	I 9 I

Figure 12-1. MEMLOGAN Output

ERROR MESSAGES

When MEMLOGAN encounters an operational error, it displays information describing the error and terminates. If the error was detected by the MPE File Management System for a file maintained by MEMLOGAN, the information displayed consists of the File Information Display (described in *MPE Intrinsic Reference Manual*), followed by one of the messages:

- * OUT FILE ERROR * For errors encountered while writing output display.
- * LOG FILE ERROR * For errors encountered while reading the MEMLOG file.

If the error was detected by MEMLOGAN rather than by the MPE File Management System, MEMLOGAN prints a message in the following format:

* MEMLOGAN ERROR: *errnum* *

In this message, *errnum* is one of the digits 1 or 2:

<i>errnum</i>	<i>Meaning</i>
1	MEMLOGAN could not lock (FLOCK) the MEMLOG file for exclusive use.
2	MEMLOGAN could not unlock (FUNLOCK) the MEMLOG file to release it for access by other processes.

MEMTIMER allows you to change the time-interval between successive updates of the memory log file MEMLOG.PUB.SYS by the system process MEMLOGP.

NOTE

No special MPE capabilities are required to run this program.

Typically, the 60-minute logging interval assigned by default each time the system is initialized is sufficient to provide the average installation with an adequate overview of system memory performance. However, in certain instances, you may wish to request more frequent memory monitoring. To initiate this change, run MEMTIMER by logging-on to the system and entering:

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

In this command, *n* is any integer greater than zero but less than 65536 that denotes the new logging interval in seconds. As an example, to change the interval to 10 seconds, enter:

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

MEMTIMER immediately alters the logging interval and then terminates the current interval period. This causes MEMLOGP to update the log file immediately and then periodically thereafter according to the new interval.

To restore the default logging interval of 60 minutes (3600 seconds), enter:

```
:RUN MEMTIMER.PUB.SYS; PARM=3600
```

NOTE

Each time the system is initialized, the default logging interval becomes the current logging interval. Thus, to maintain any interval other than the default interval, you must run MEMTIMER with *n* set to the appropriate value after each system cold-load.

When an error occurs while MEMTIMER is running, one of the messages shown in Table 13-1 may appear:

Table 13-1. MEMTIMER Error Messages

Message	Meaning/Action
** INVALID PARM (DELAY) VALUE **	You entered invalid value for <i>n</i> in :RUN command to execute MEMTIMER. Current logging interval remains unchanged and MEMTIMER terminates. Re-run MEMTIMER with appropriate <i>n</i> value.
** MEMORY LOGGING PROCESS NOT ACTIVE **	MEMLOGP was terminated before you requested MEMTIMER. (This termination may occur, for instance, if a File Management System error occurs when MEMLOGP attempts to access the log file.) MEMTIMER terminates. When MEMLOGP is active again, re-run MEMTIMER.
** MEMLOGP TIMER ENTRY NOT FOUND **	Either memory-logging hardware is absent from system or MEMLOGP is currently updating log file. Logging interval is changed in either case. If memory-logging hardware exists on system, run MEMTIMER again to ensure that MEMLOGP recognizes new interval immediately.

A

Access-Mode Abbreviations, 8-8
ALTER Function, 11-5, 11-6
Attributes, 8-1

C

Capability-Set Abbreviations, 8-8
CIR Contents, 3-2
Cold Loading SADUTIL, 1-6
COND Function, 1-11
CONF Function, 1-13

D

DEBUG Function, 11-7
Defective Track Table, 1-17
Disc Files
 Modifying and Listing, 6-1
 Retrieving and Copying, 1-25
 Recreating Disc Files, 2-1
DISKED2, 6-1—6-3
DISKED2 Error Messages, 6-3
DPAN2, 3-1—3-3

E

EDIT Function, 1-15
EOF Mark, 1-26
EXIT Function, 11-9

F

Files
 Modifying and Listing Disc Files, 6-1
 Program Files, 9-1
 Recreating Disc Files, 2-1
 Retrieving and Copying Disc Files, 1-25
 Segmented Library (SL) Files, 10-1
FIND Function, 11-11, 11-12
FREE2, 7-1, 7-2
Free-Space Table, 7-1

I

INPUT Function, 11-13, 11-14

L

LIST Function, 11-15, 11-16
LISTDIR2, 8-1—8-10
LISTDIR2 Command Examples, 8-5
LISTEQ2, 5-1
LISTLOG2, 4-1
LOGLIST, 4-1

M

MANAGER SYS, 4-1—4-3
MDUMP. PUB. SYS, 3-2
MEMLOGAN
 Error Messages, 12-5
 Memory-Logging Operations, 12-1
 Output, 12-2
 Running, 12-1
Memory Dump, 3-1
MEMTIMER, 13-1, 13-2
MODE Function, 11-17

O

OUTM Function, 1-17
OUTPUT Function, 11-19, 11-20

P

Passwords, 8-10
PATCH, 9-1—9-3
PATCH Error Numbers, 9-3
PDSK Function, 1-19
PDTT Function, 1-21
PFIL Function, 1-23
PFRE Function, 1-25
Program Files, 9-1
PURGE Function, 11-21
PVOL Function, 1-27

R

RECOVER2, 2-1, 2-2

S

SADUTIL, 1-1
 Creating a Cold-Load Tape, 1-2—1-5
 Dialogue With Operator, 1-7
 Error Messages, 1-34, 1-35
SAVE Function, 1-29
Security Restrictions, 8-9
Segmented Library (SL) Files, 10-1
SHOW Function, 11-23—11-25
SLPATCH, 10-1—10-3
SPOOK
 Command Entry, 11-3
 Commands, 11-5—11-29
 Error Messages, 11-31
 Operations, 11-1, 11-2
 Running, 11-2
STOP Function, 1-33



T

TEXT Function, 11-27

U

User-Type Abbreviations, 8-9

X

XPLAIN Function, 11-29

Filing No. 3MPE.320.30000-90044
Part No. 30000-90044
Printed in U.S.A.

3/77



Sales and service from 172 offices in 65 countries.
5303 Stevens Creek Blvd., Santa Clara, California 95050